Production Hardening: Customizing Towards Reliable Data

Justin Pihony

DEVELOPER SUPPORT MANAGER @ LIGHTBEND

@JustinPihony

Hardening & Customizing

More Configuration

- Producer
- Consumer

Reliability

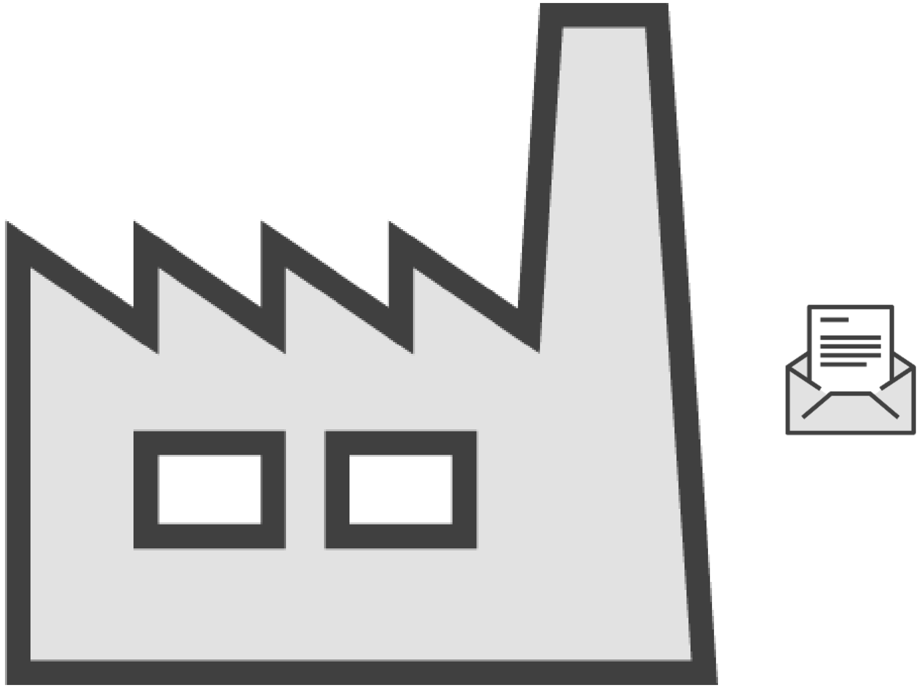
Message Customization

Configuring the Producer

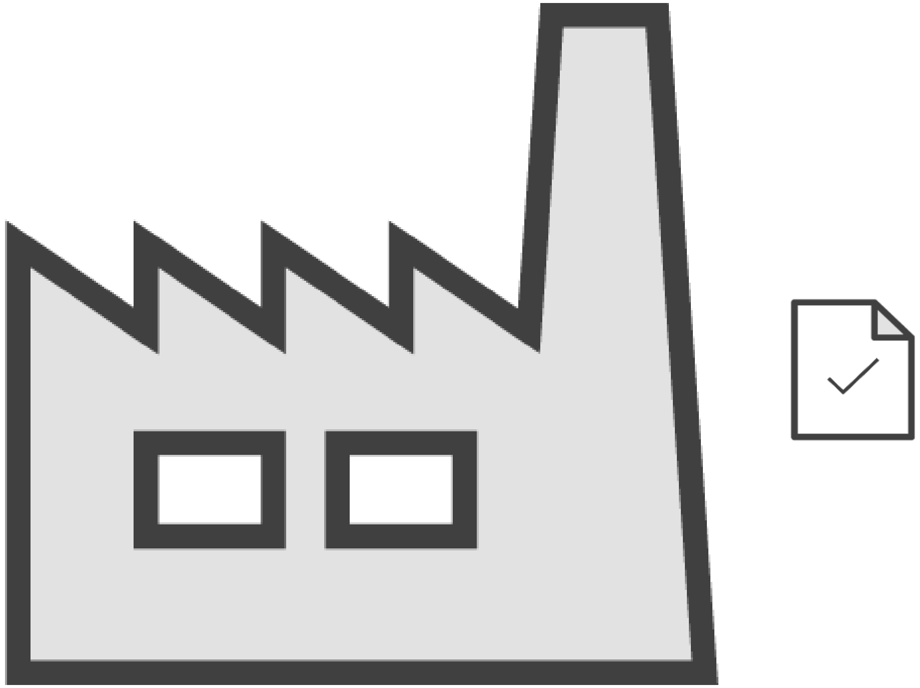
Producer

Producer

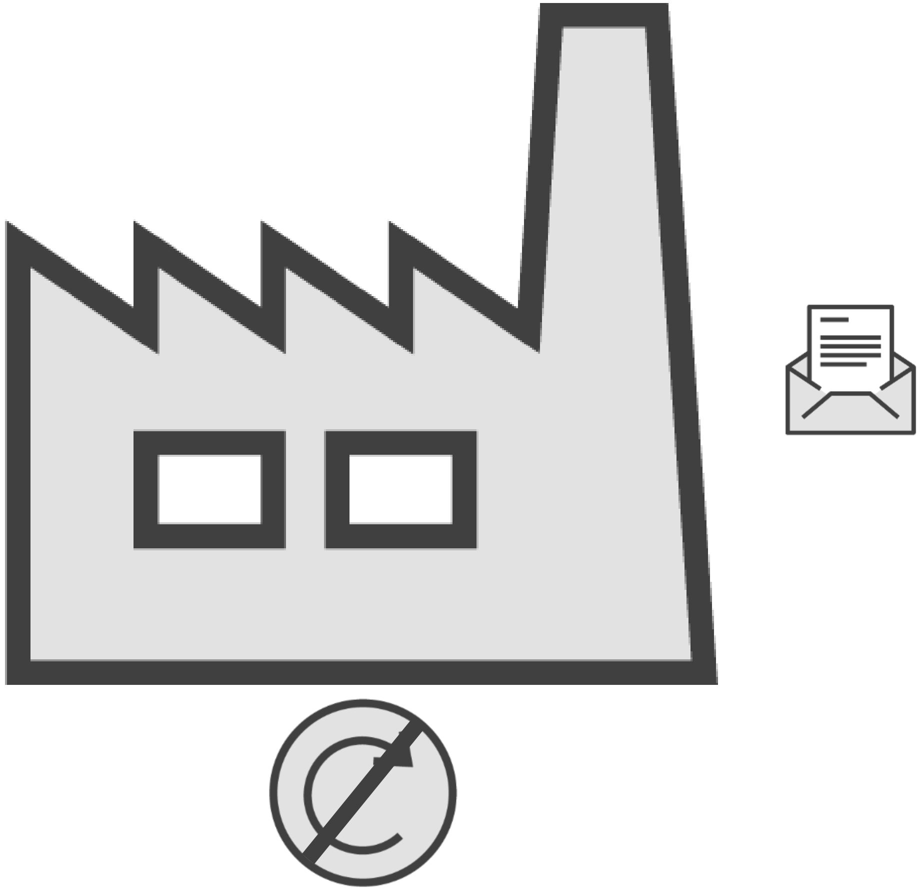
acks



acks

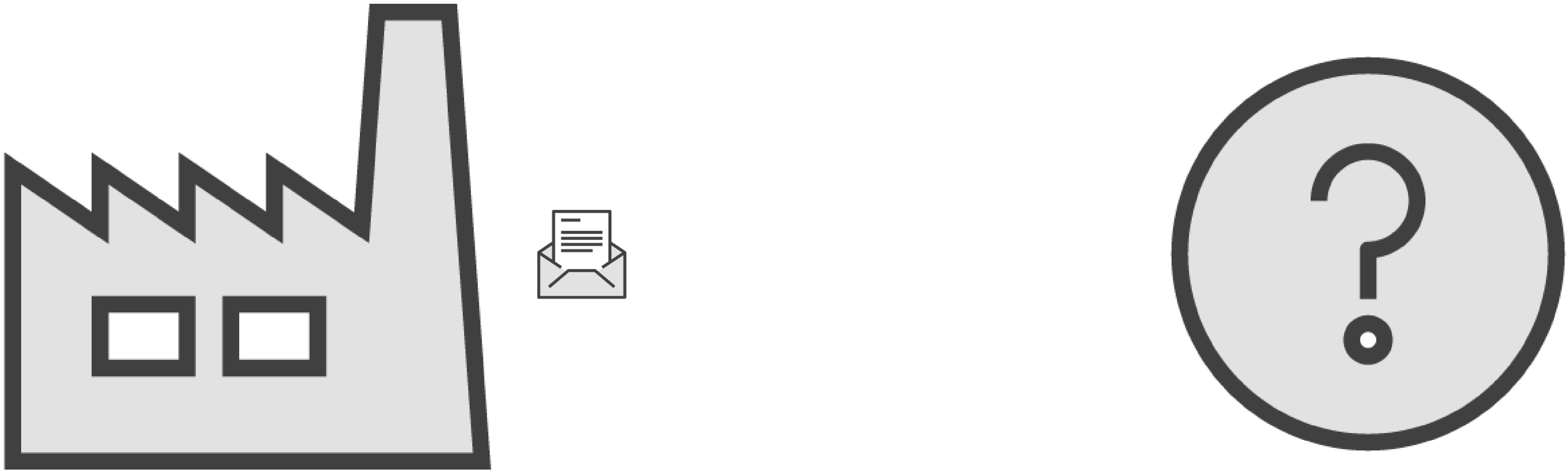


acks = 0

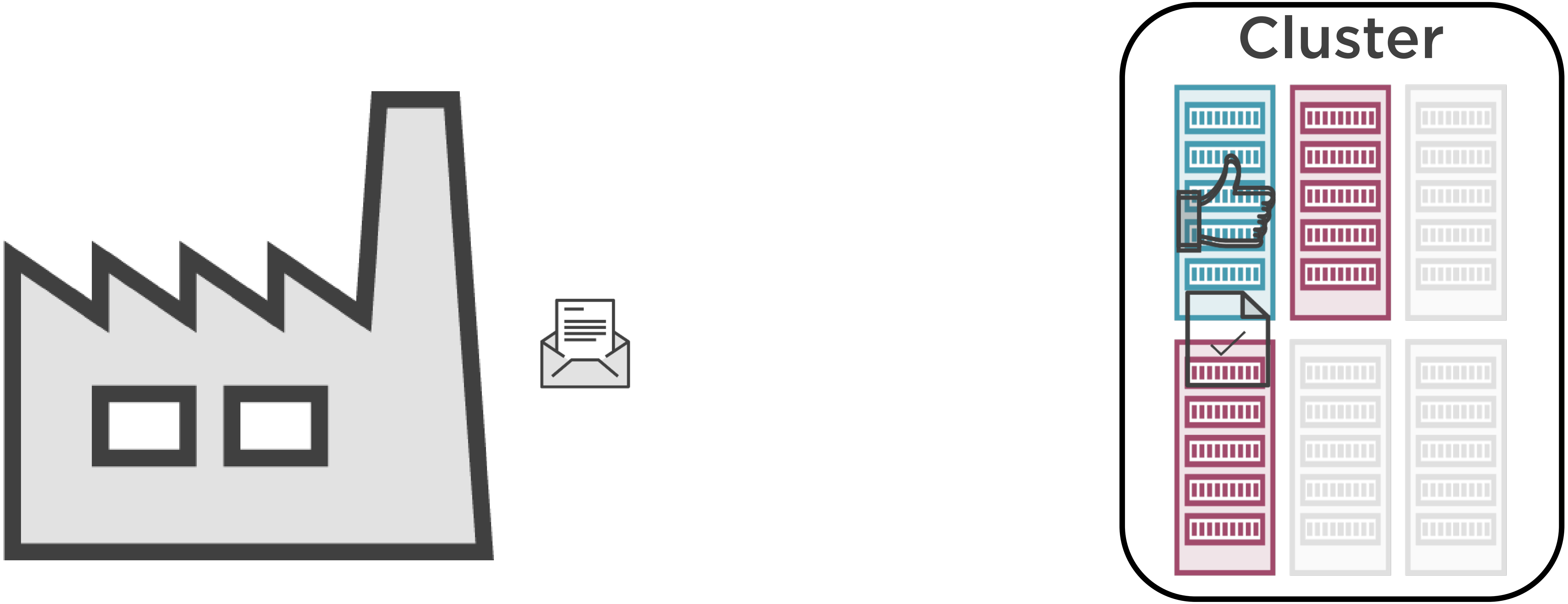


`recordMetadata.offset = -1`

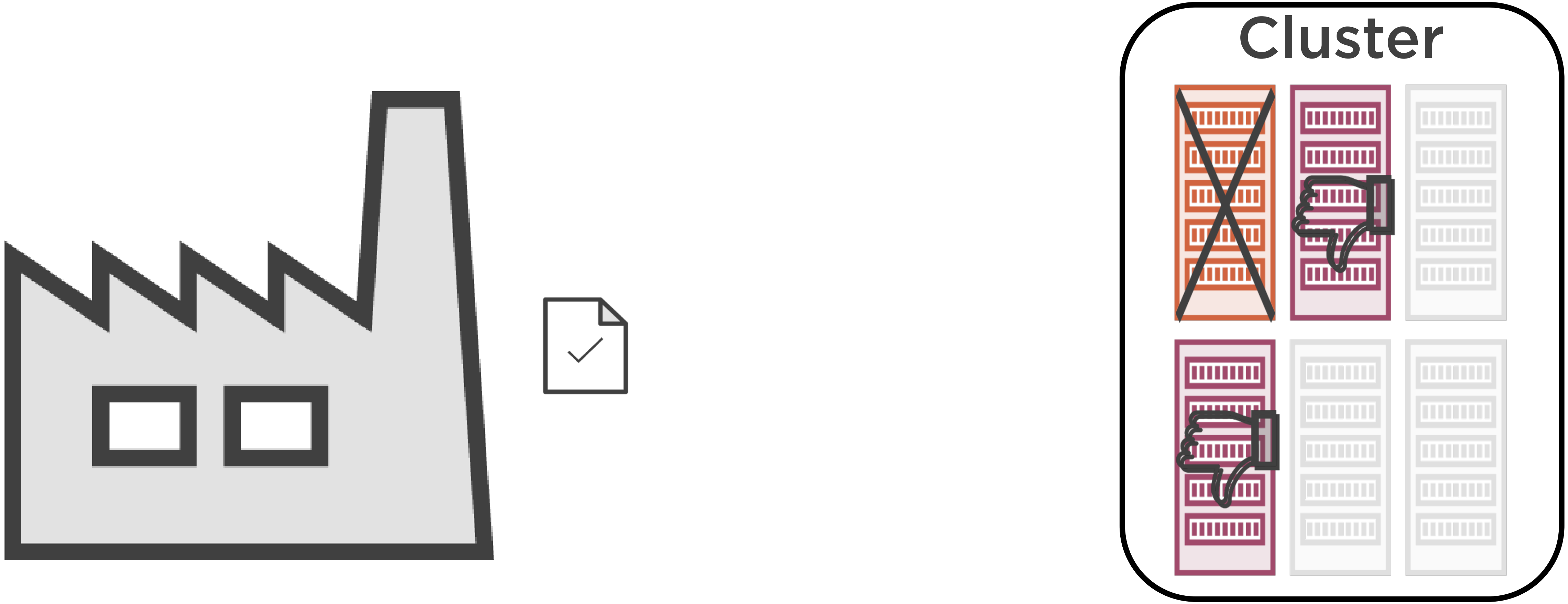
acks = 0



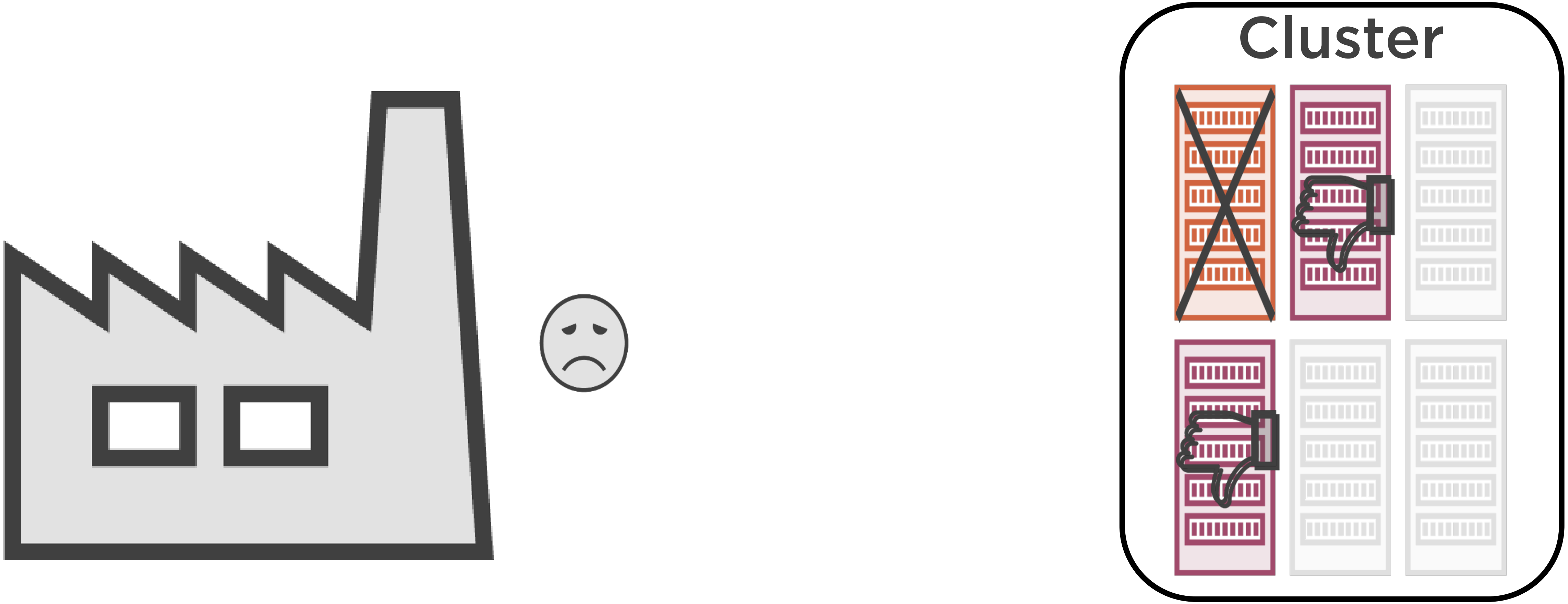
acks = 1 (default)



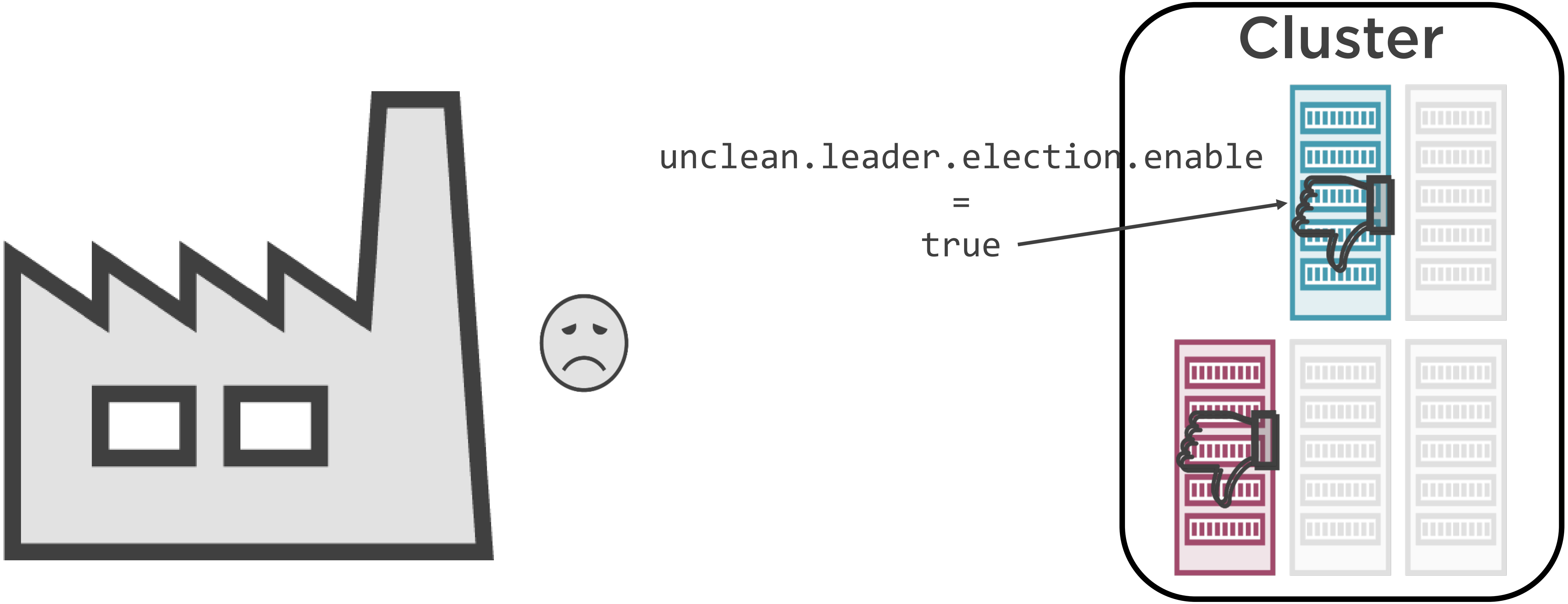
acks = 1 (default)



acks = 1 (default)

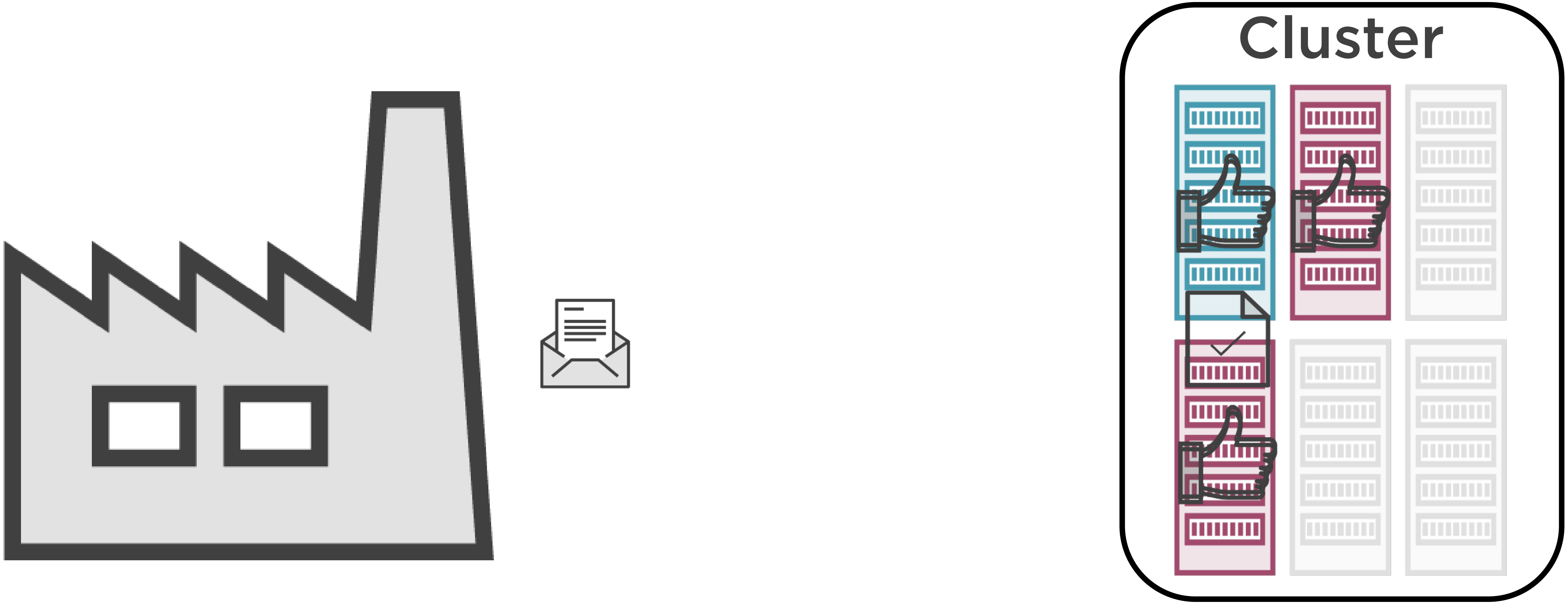


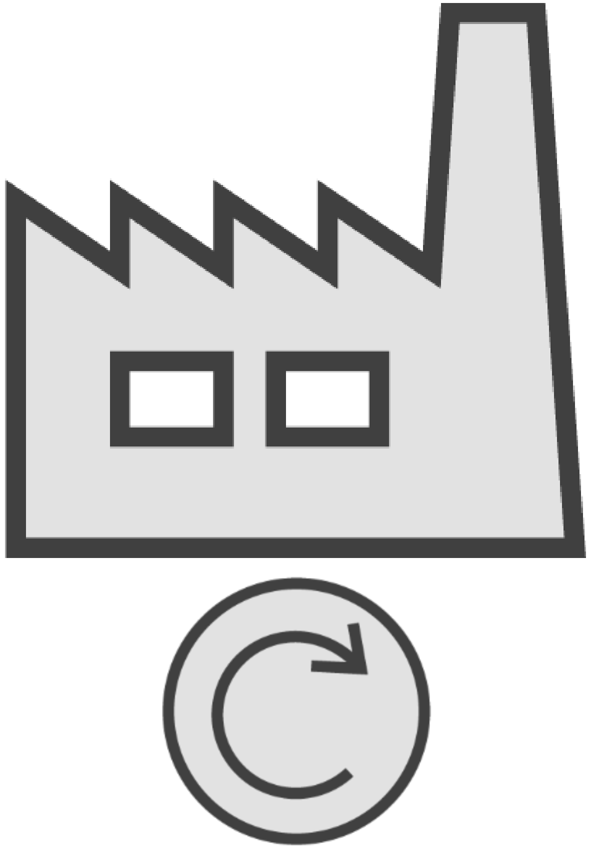
acks = 1 (default)



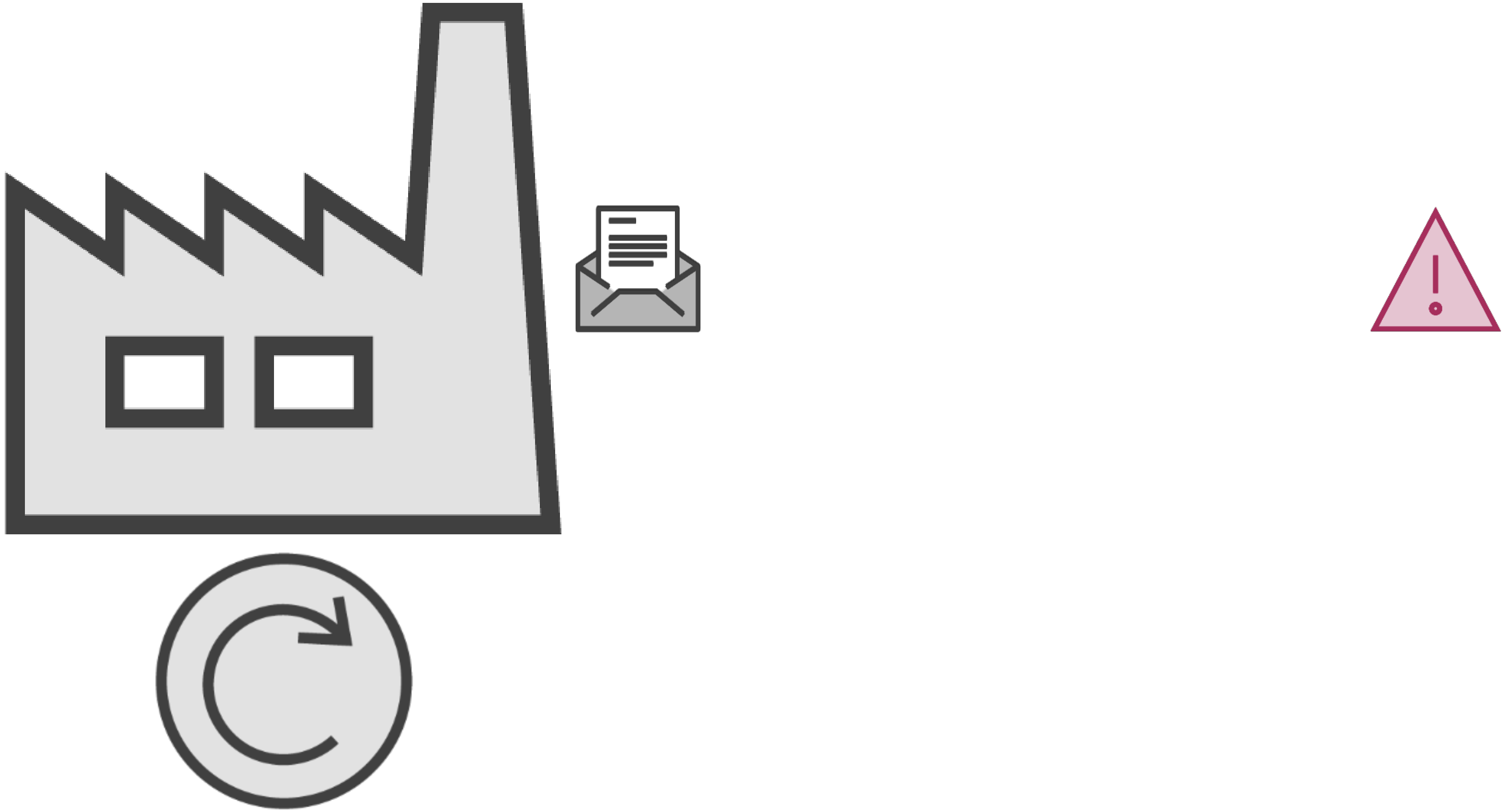
unclean.leader.election.enable
=
true

acks = all (or -1)

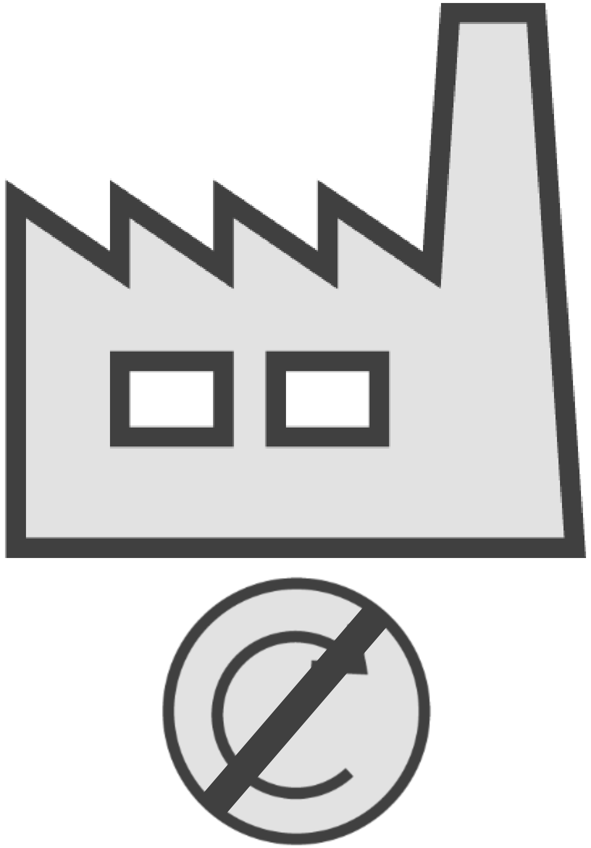




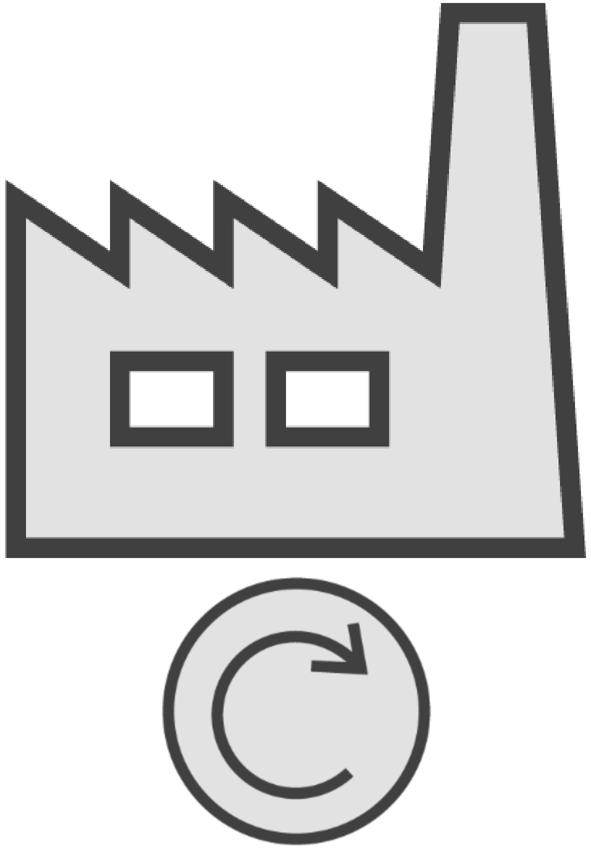
retries



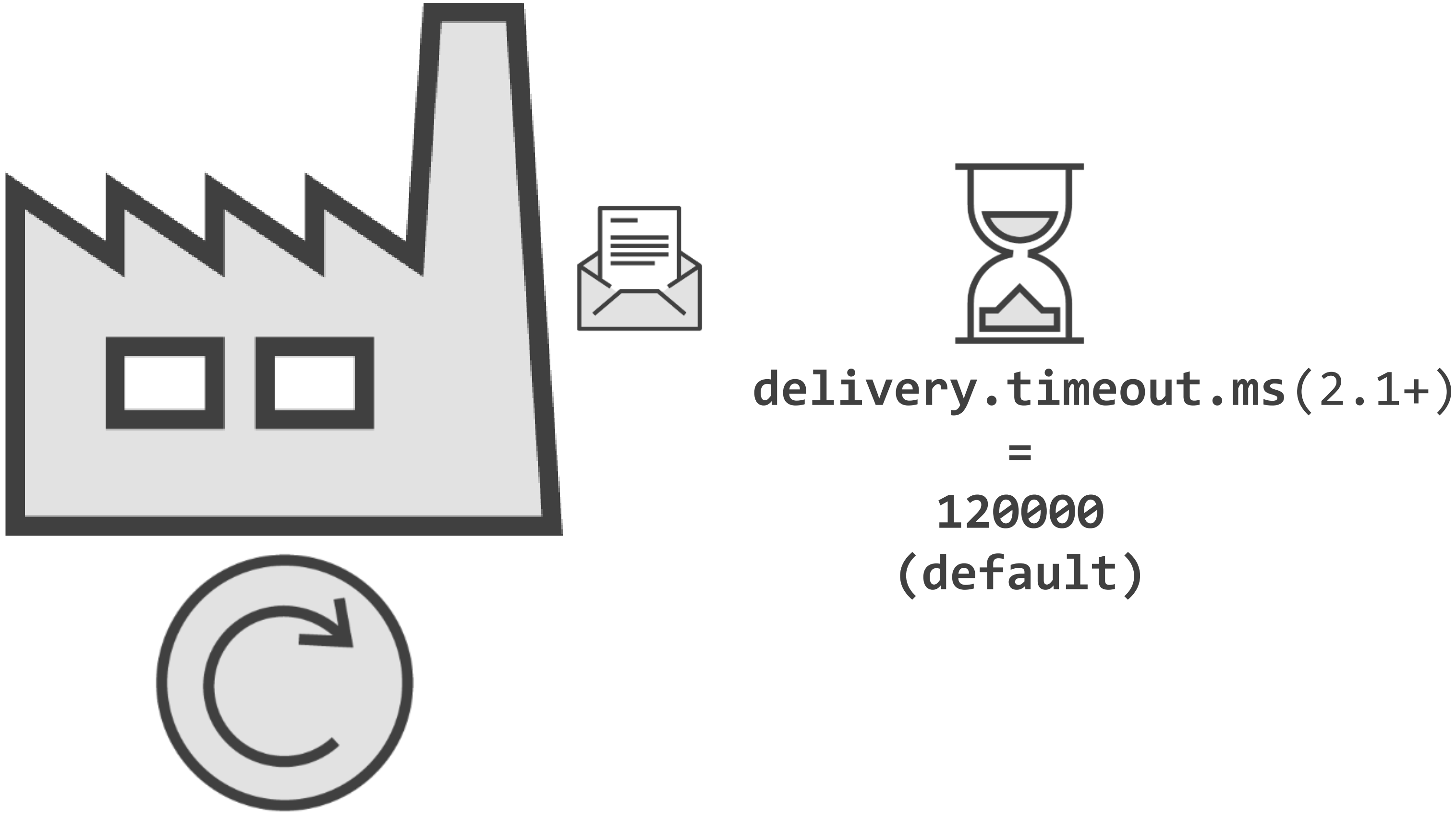
retries



retries = 0 (off)



retries = 0 (off)



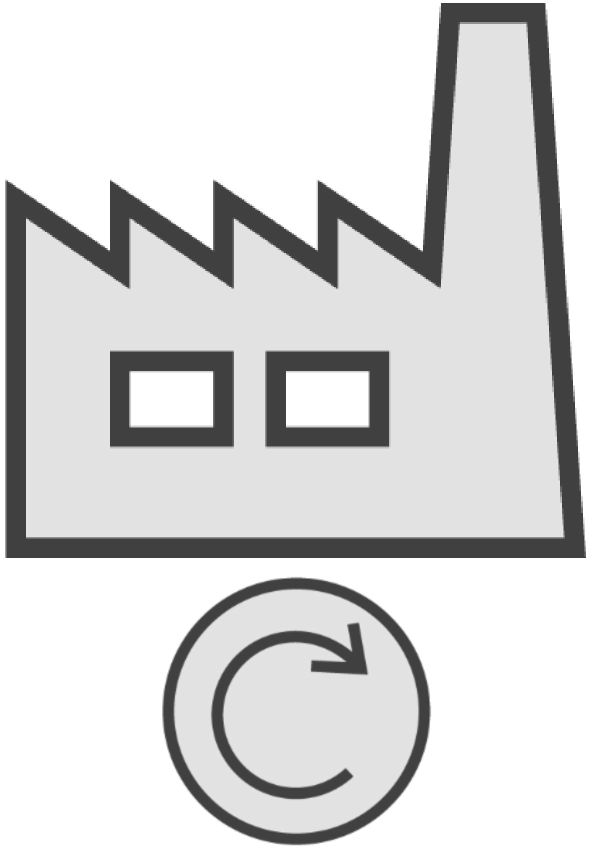
delivery.timeout.ms(2.1+)

=

120000

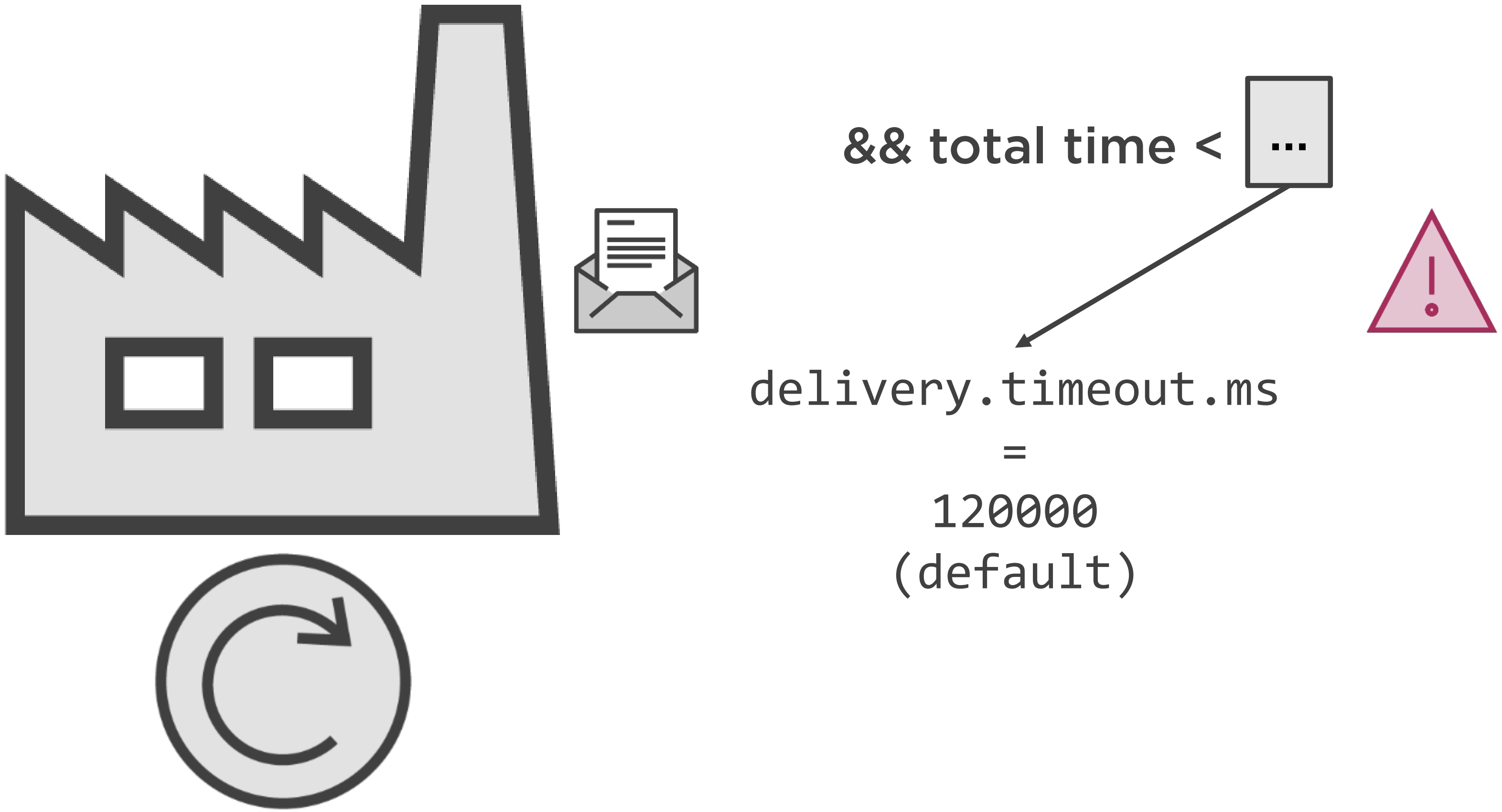
(default)

retries = Integer.MAX_VALUE
(default in 2.1+)



delivery.timeout.ms
=
120000
(default)

retries = 1

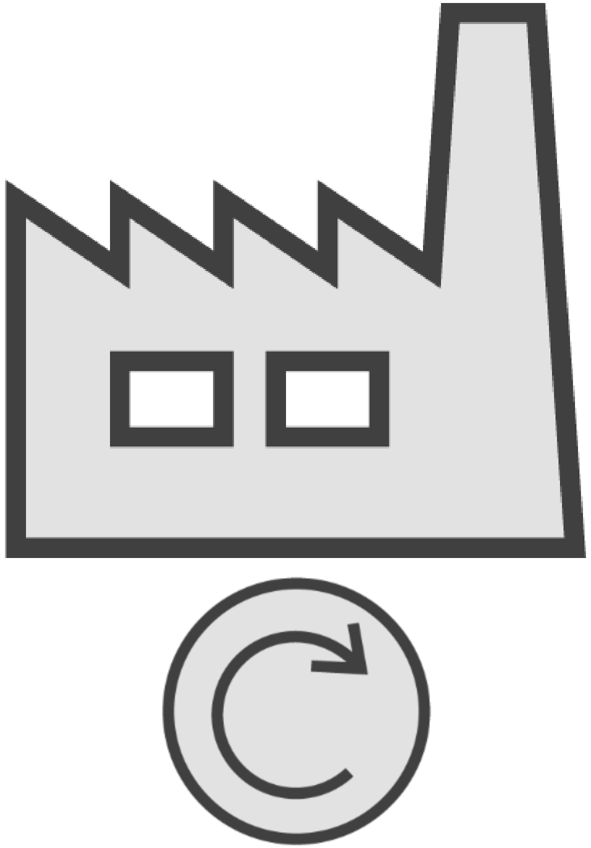


retried = 1

&& total time < ...

delivery.timeout.ms
=
120000
(default)

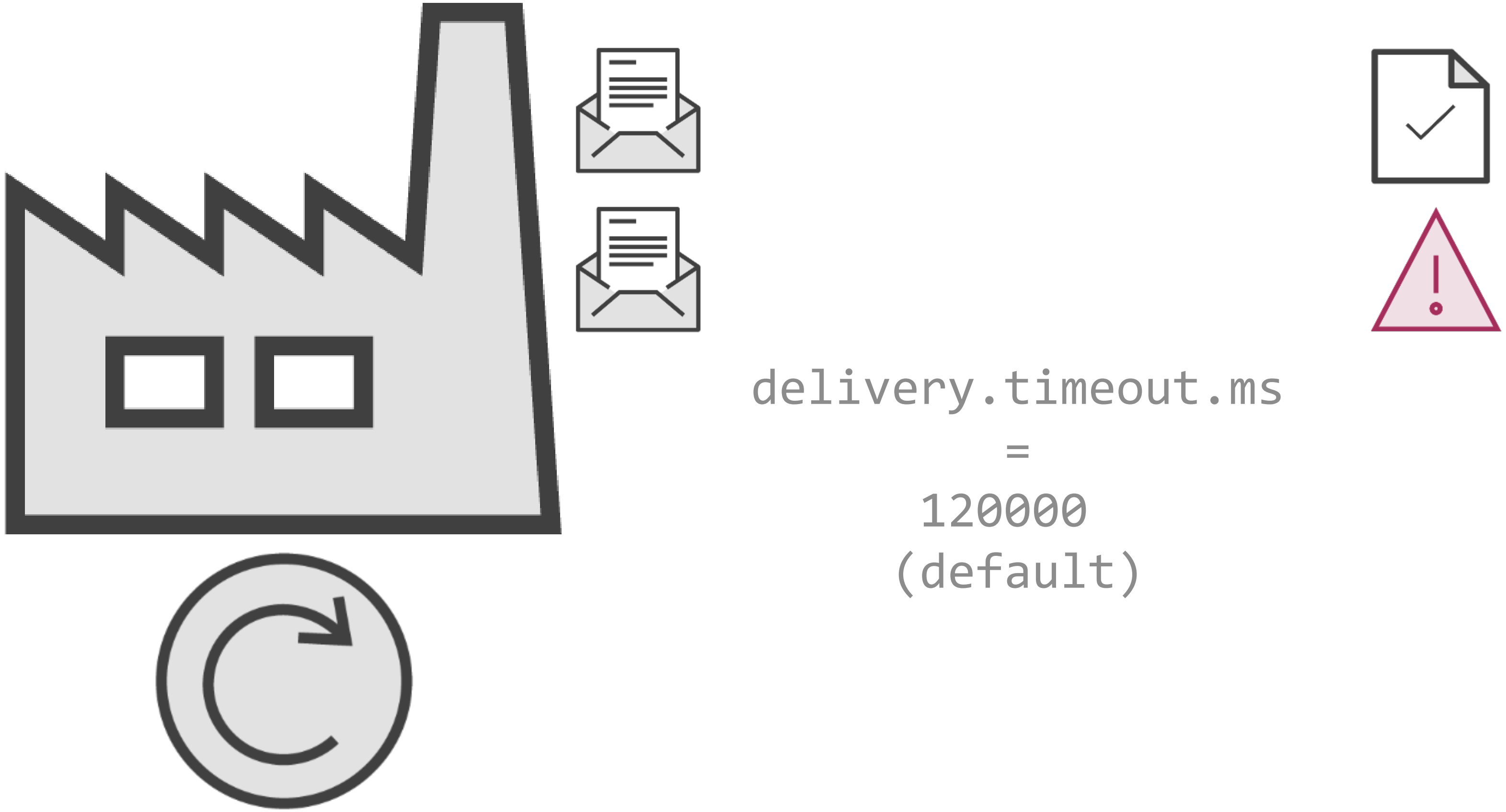
retries = 1



delivery.timeout.ms
=
120000
(default)

retry.backoff.ms = 100 (default)
(0.10.2+)

retries = 1



delivery.timeout.ms
=
120000
(default)

retry.backoff.ms = 100 (default)

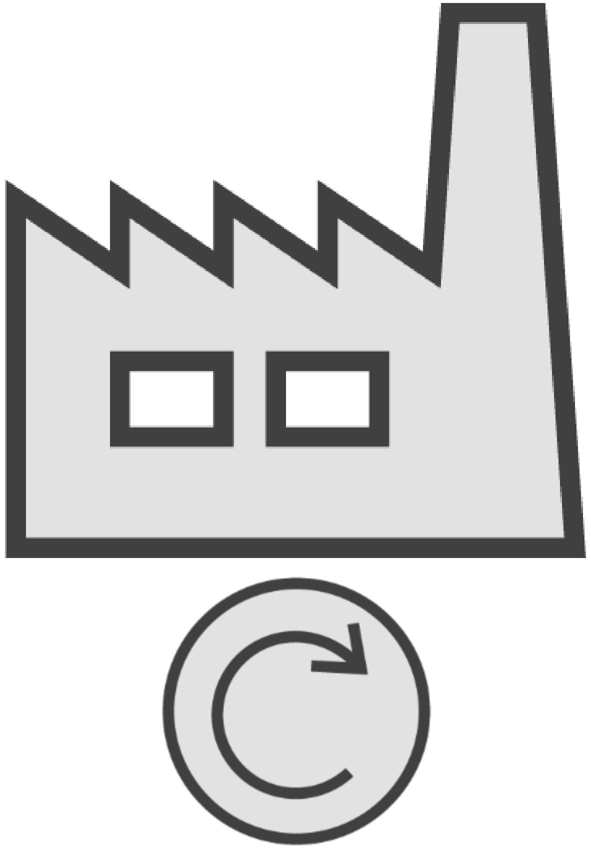
retries = 1

max.in.flight.requests.per.connection

=

5

(default)



delivery.timeout.ms

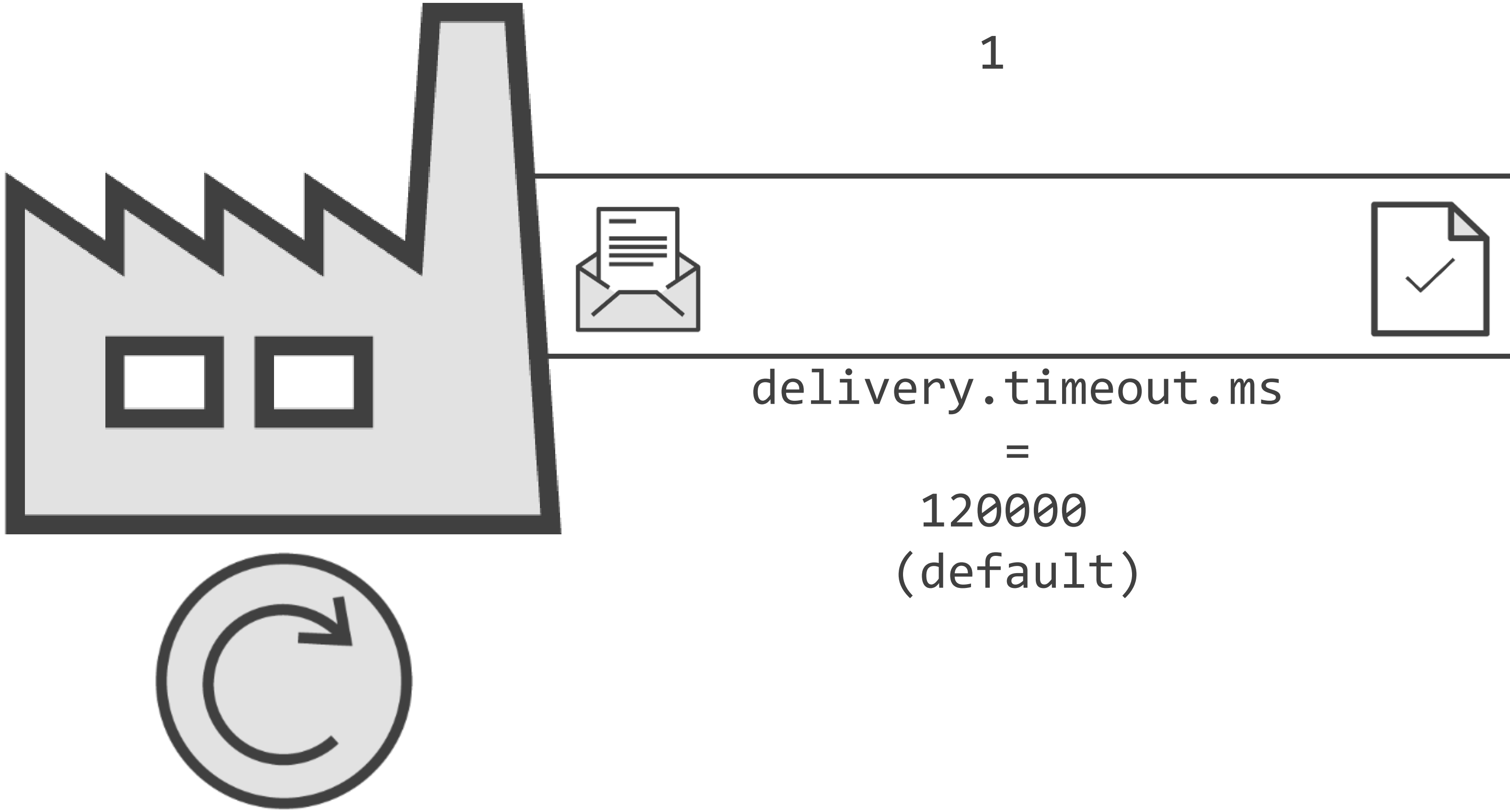
=

120000

(default)

retry.backoff.ms = 100 (default)

retries = 1



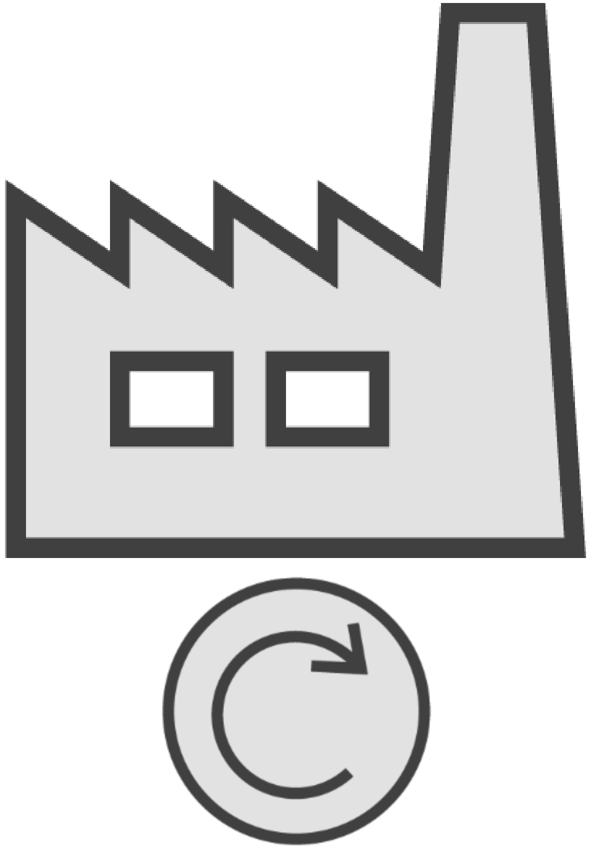
max.in.flight.requests.per.connection
=
1

delivery.timeout.ms
=
120000
(default)

retry.backoff.ms = 100 (default)

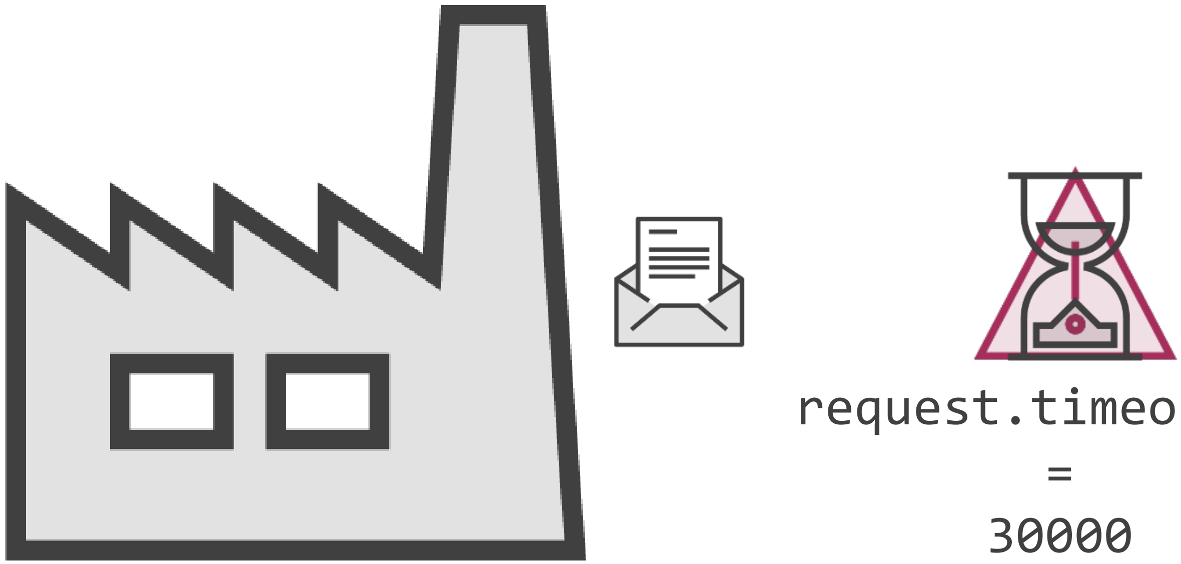
retries = 1

max.in.flight.requests.per.connection
=
5
(default)



delivery.timeout.ms
=
120000
(default)

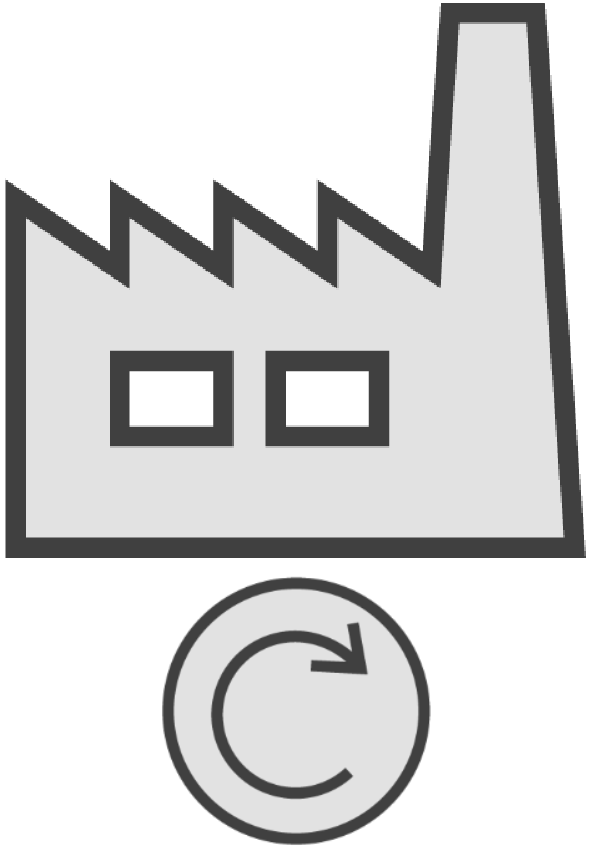
retry.backoff.ms = 100 (default)



request.timeout.ms
=
30000
(default)

retries

max.in.flight.requests.per.connection

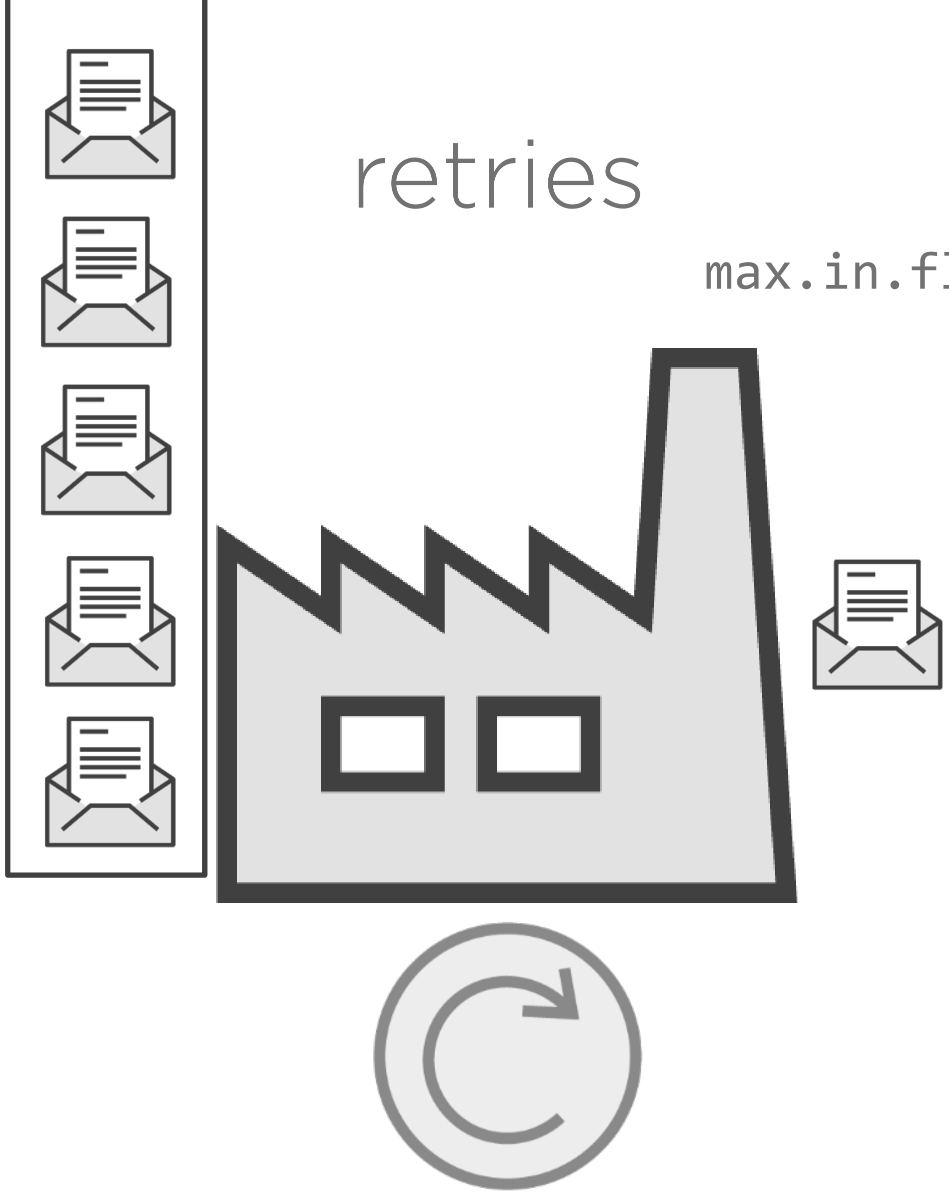


request.timeout.ms

delivery.timeout.ms

retry.backoff.ms

retries



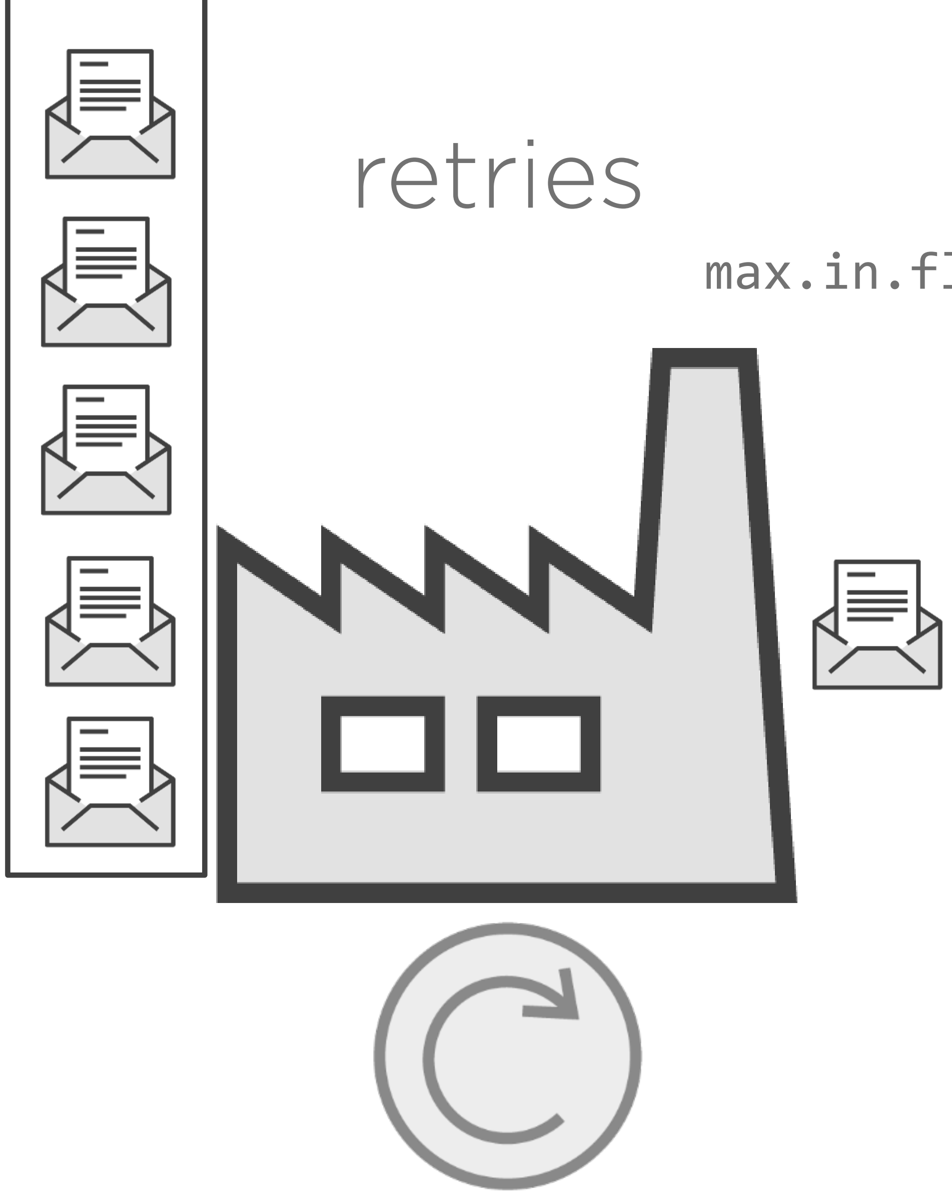
max.in.flight.requests.per.connection

buffer.memory
=
33554432
(default)

request.timeout.ms

delivery.timeout.ms

retry.backoff.ms



retries

max.in.flight.requests.per.connection

buffer.memory
=
33554432
(default)

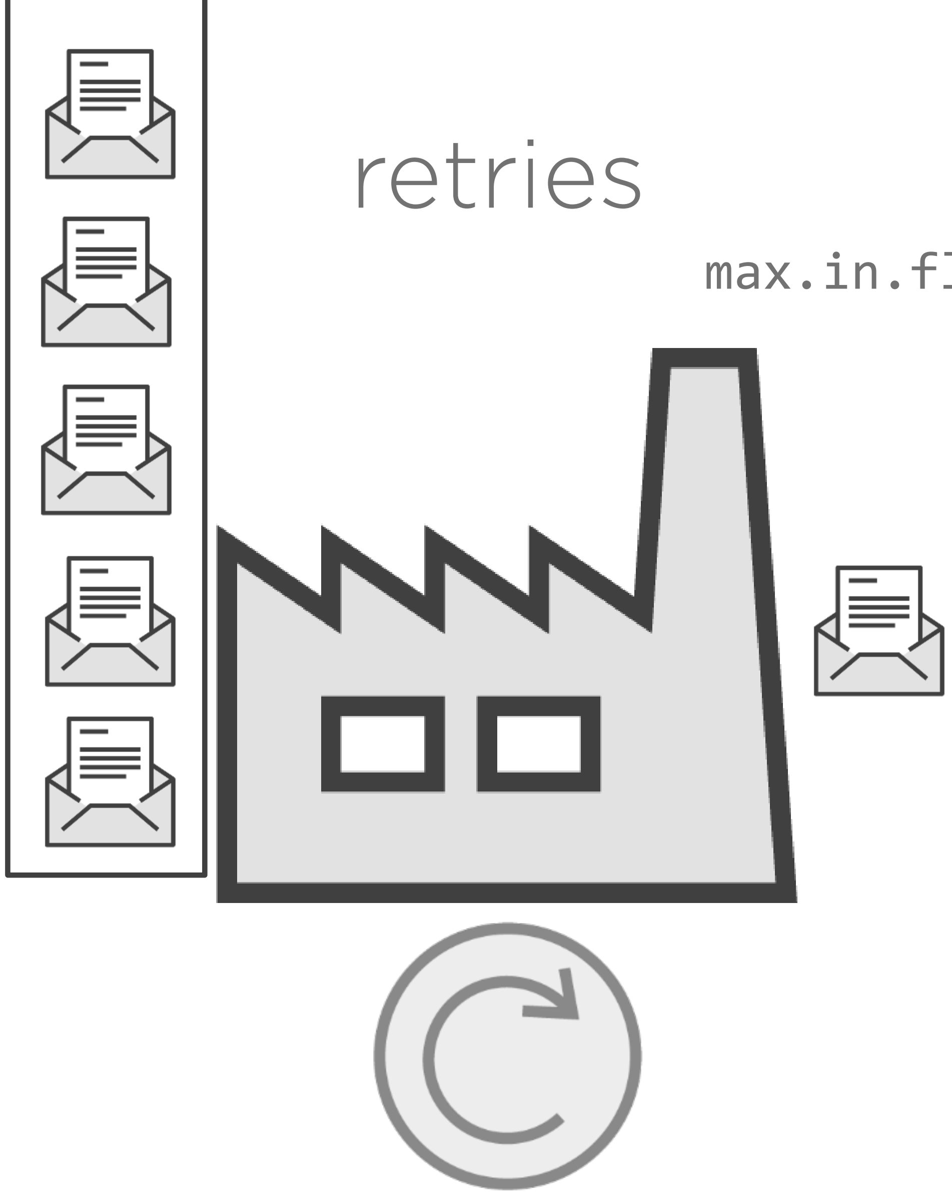
request.timeout.ms

delivery.timeout.ms

retry.backoff.ms

max.block.ms
=
60000
(default)

buffer.memory
=
33554432
(default)



retries

max.in.flight.requests.per.connection

request.timeout.ms

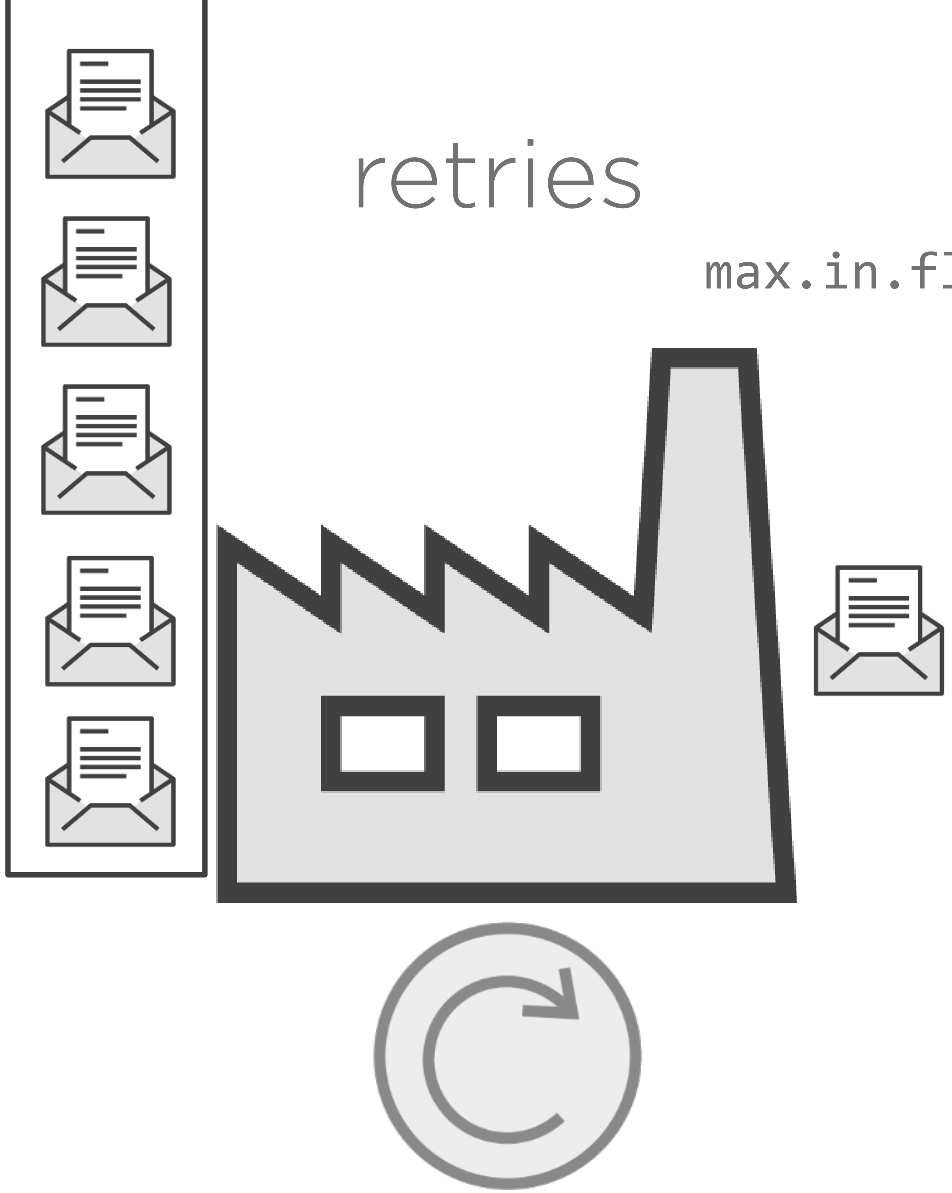
delivery.timeout.ms

retry.backoff.ms

block.on.buffer.full
(< 0.9)
=
true
(default)

max.block.ms
=
60000
(default)

buffer.memory
=
33554432
(default)



retries

max.in.flight.requests.per.connection

request.timeout.ms

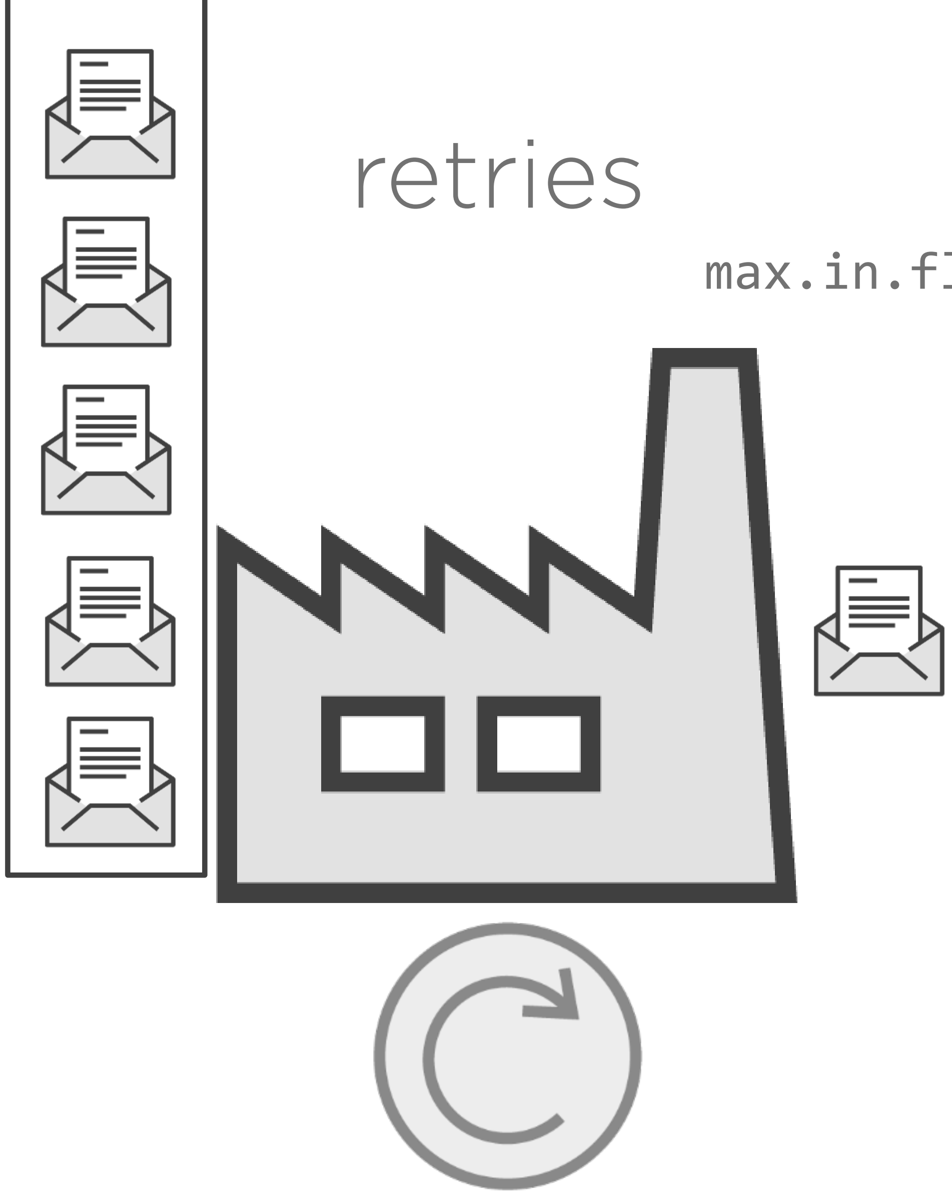
delivery.timeout.ms

retry.backoff.ms

block.on.buffer.full
=
false
(default)

max.block.ms
=
60000
(default)

buffer.memory
=
33554432
(default)



retries

max.in.flight.requests.per.connection

request.timeout.ms

delivery.timeout.ms

retry.backoff.ms

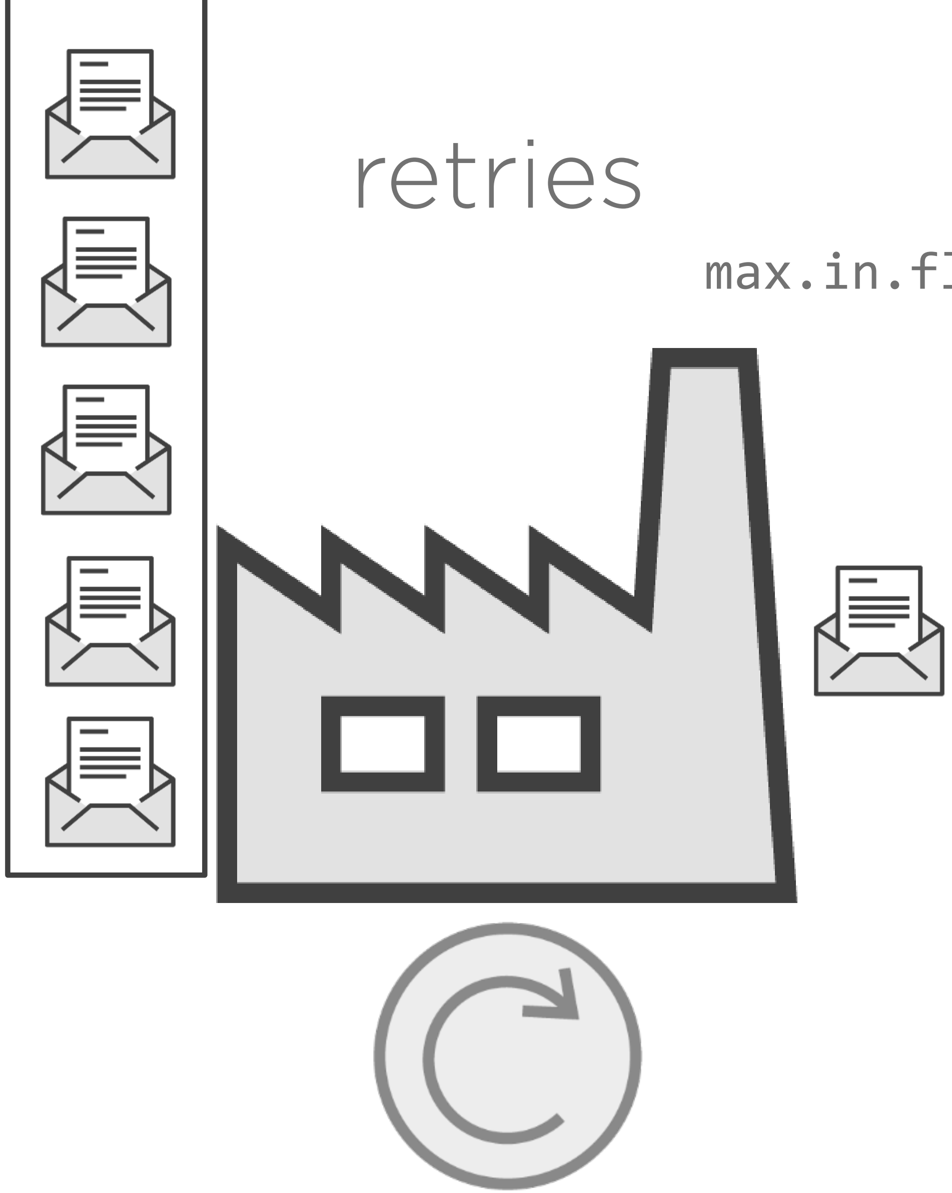
max.block.ms
=
60000
(default)

---

buffer.memory
=
33554432
(default)

retries

max.in.flight.requests.per.connection

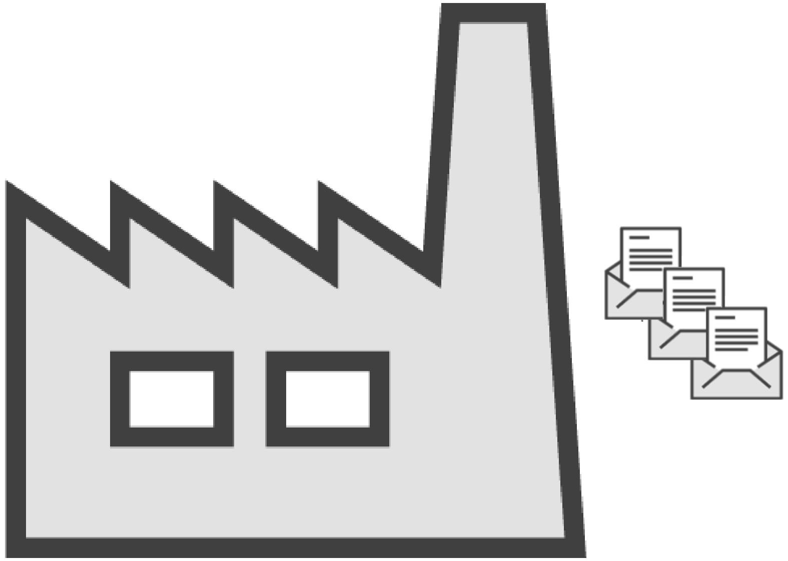


request.timeout.ms

---

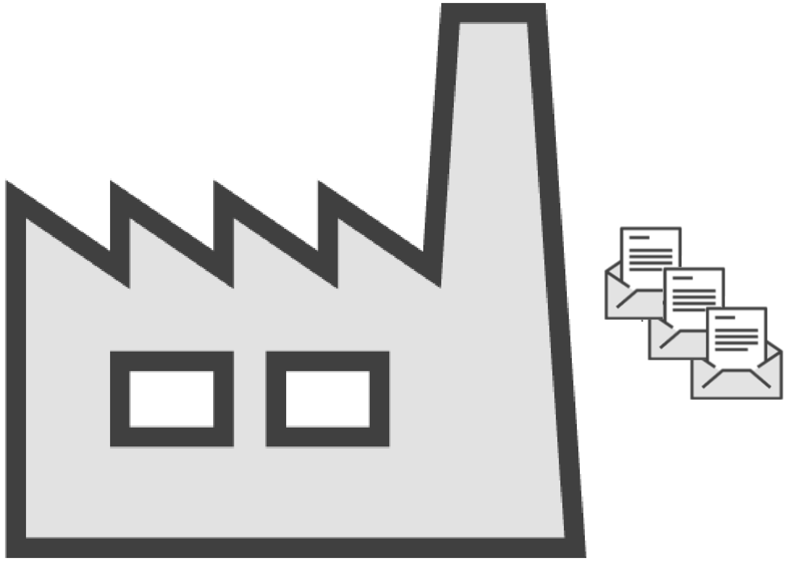
delivery.timeout.ms

retry.backoff.ms



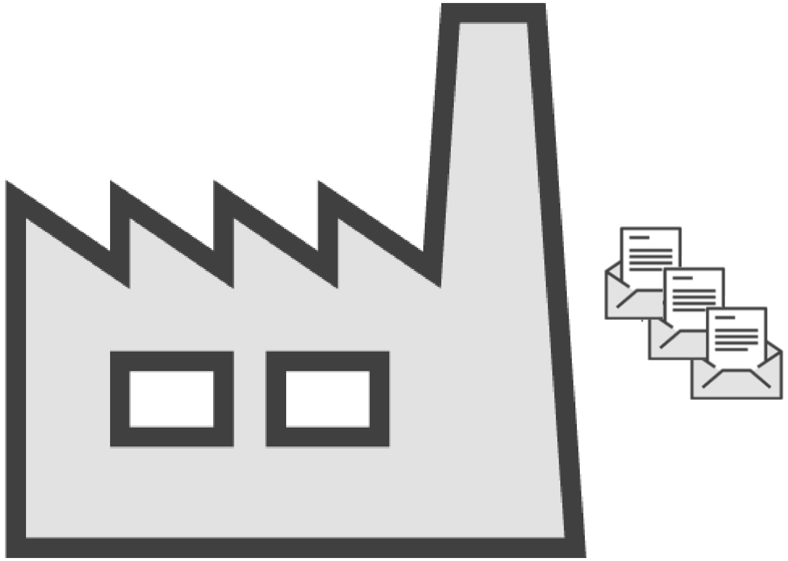
batch.size

linger.ms



~~batch.size~~|max.request.size

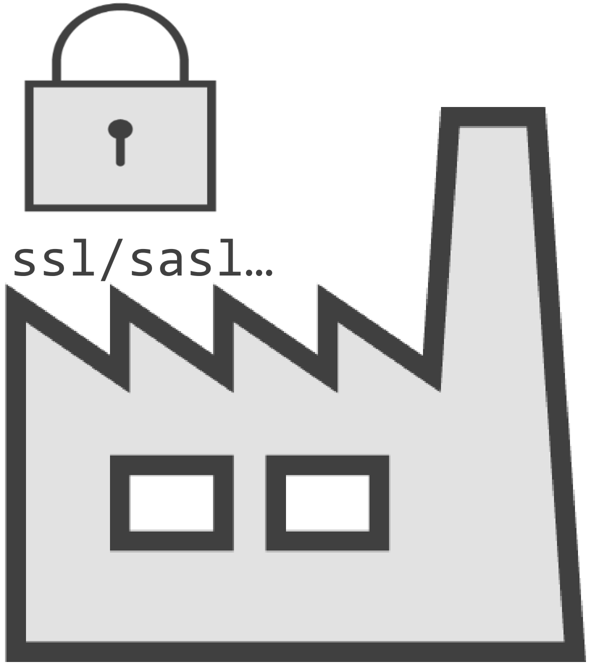
linger.ms



(broker config)
message.max.bytes
>
max.request.size

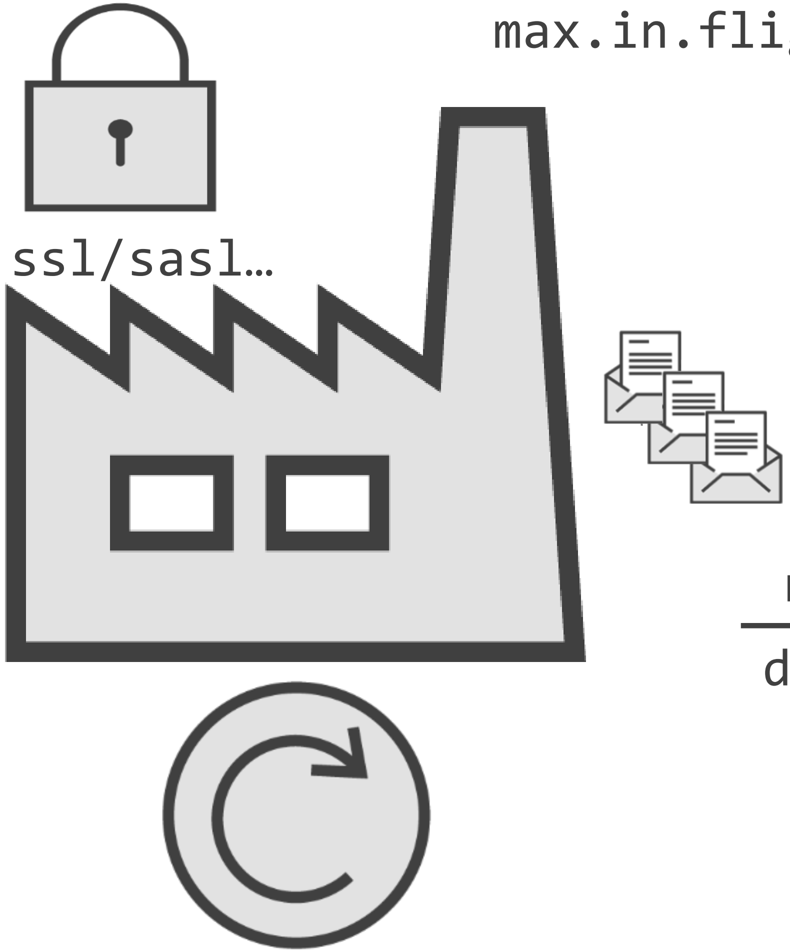
batch.size

linger.ms



ssl/sasl…

retries



max.block.ms

buffer.memory

ssl/sasl…

max.in.flight.requests.per.connection

batch.size | max.request.size

linger.ms

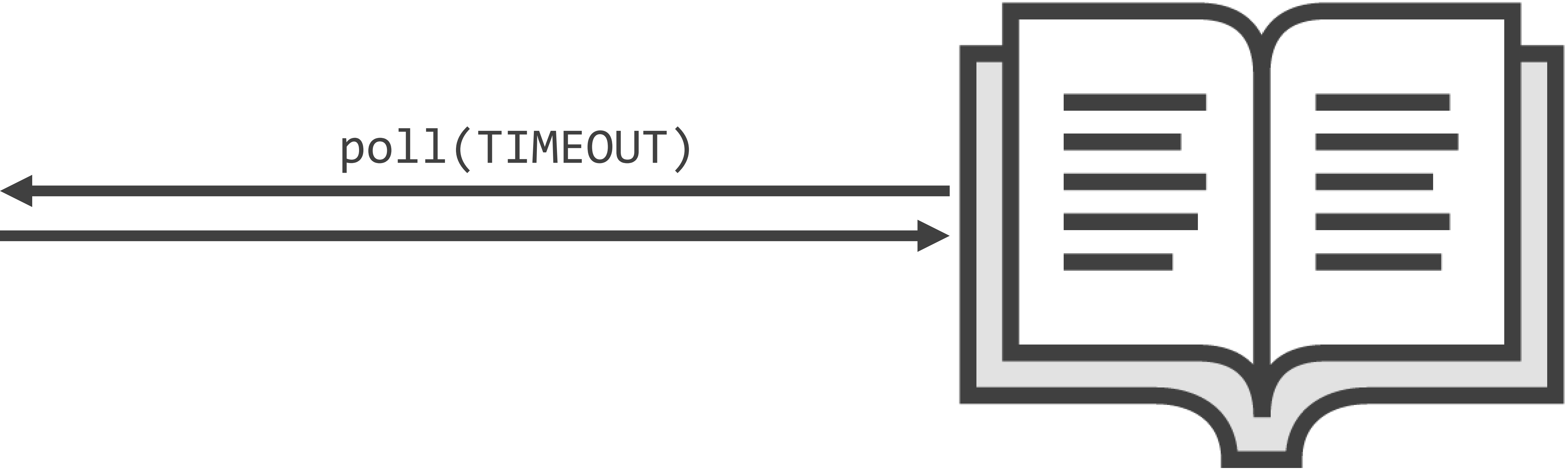
request.timeout.ms

delivery.timeout.ms

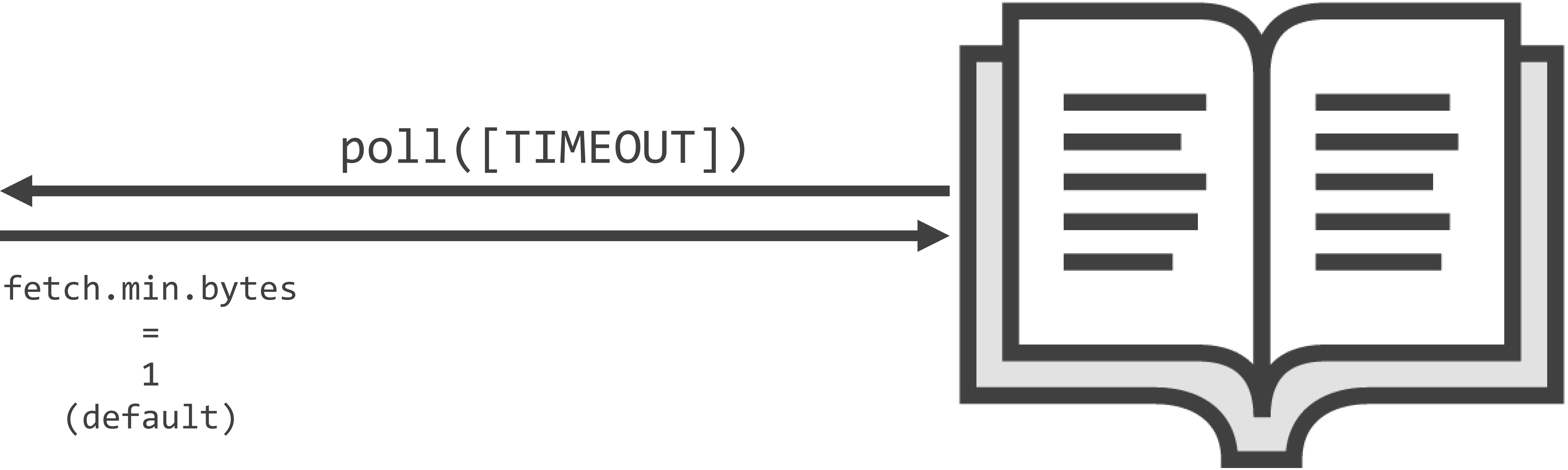
retry.backoff.ms

Configuring the Consumer

Consumer



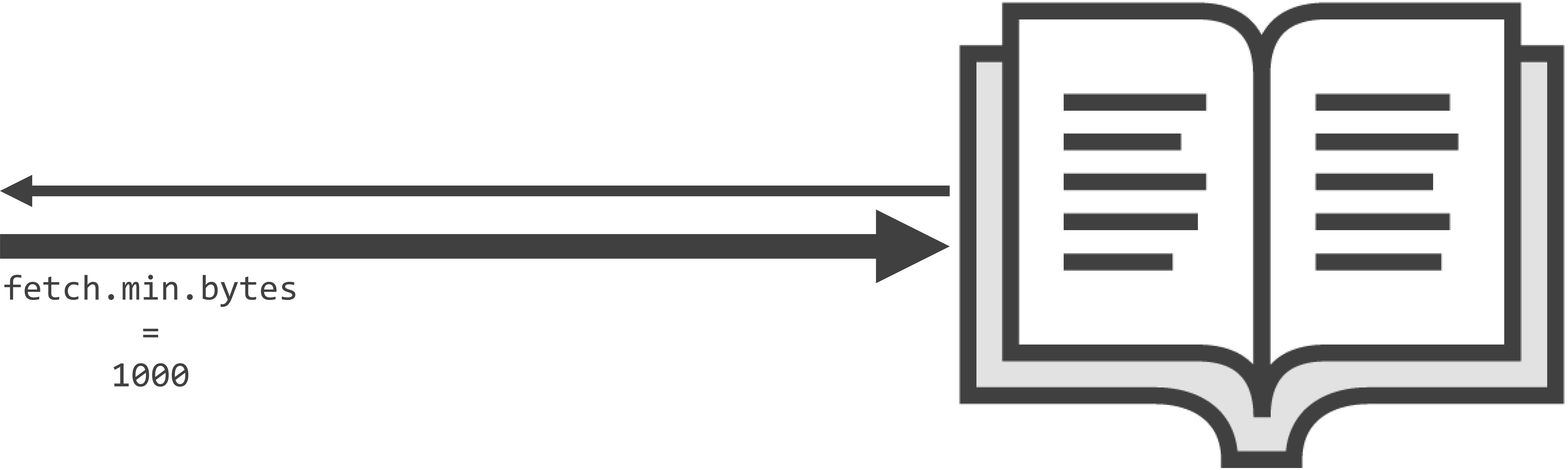
Consumer



poll([TIMEOUT])

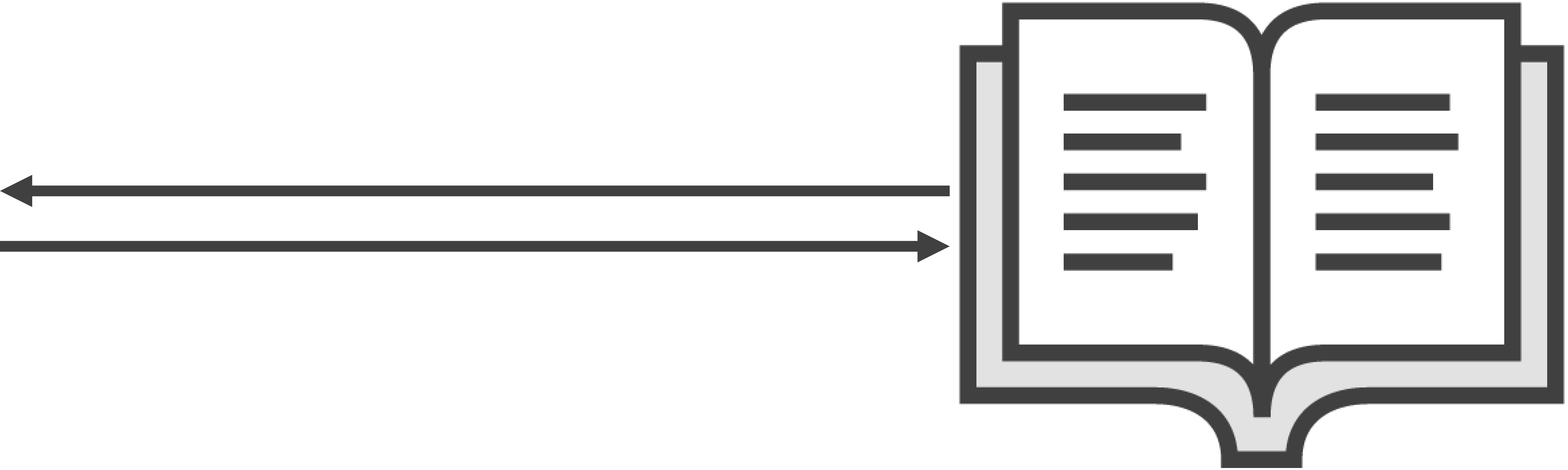
fetch.min.bytes
=
1
(default)

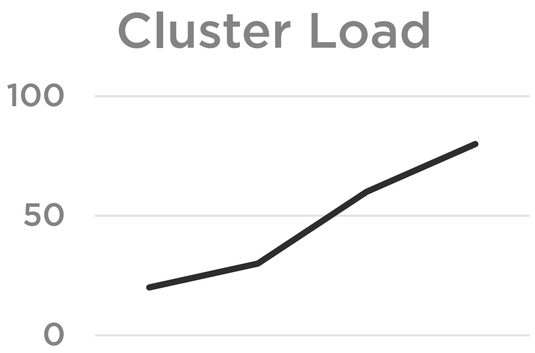
Consumer

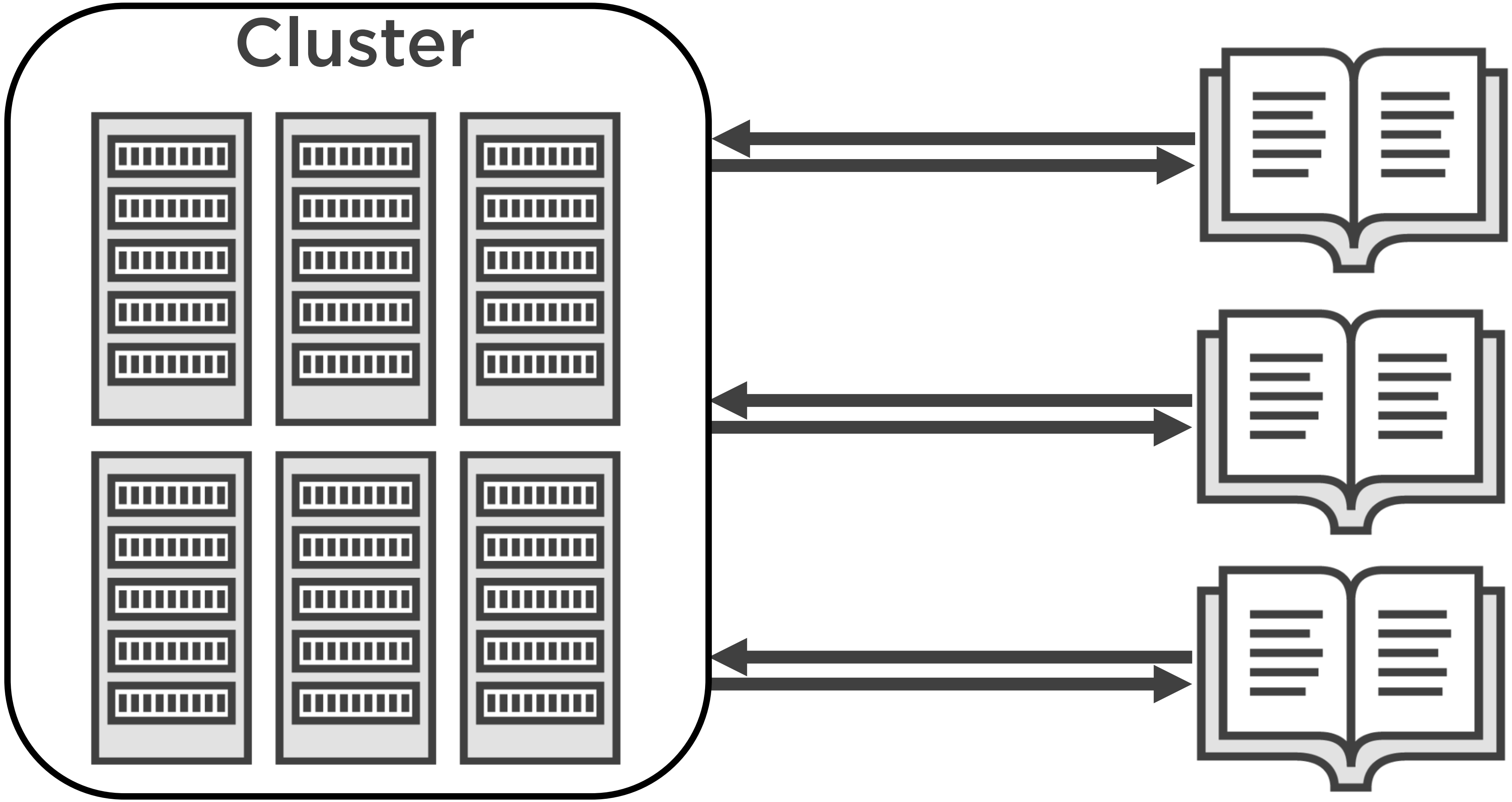


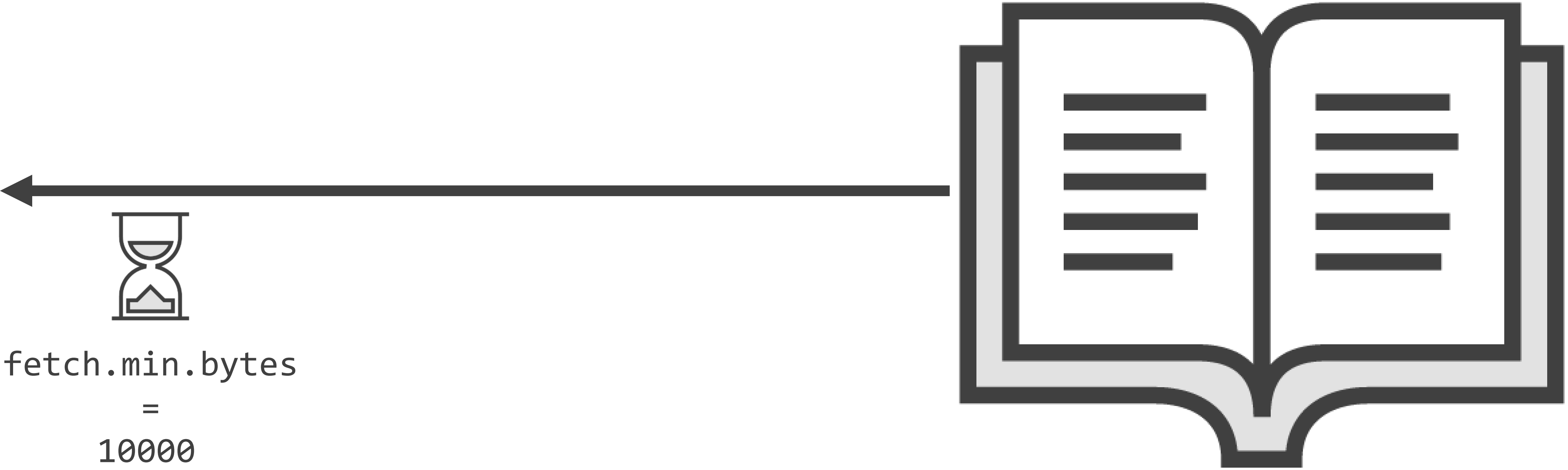
```
fetch.min.bytes
       =
     1000
```

Consumer

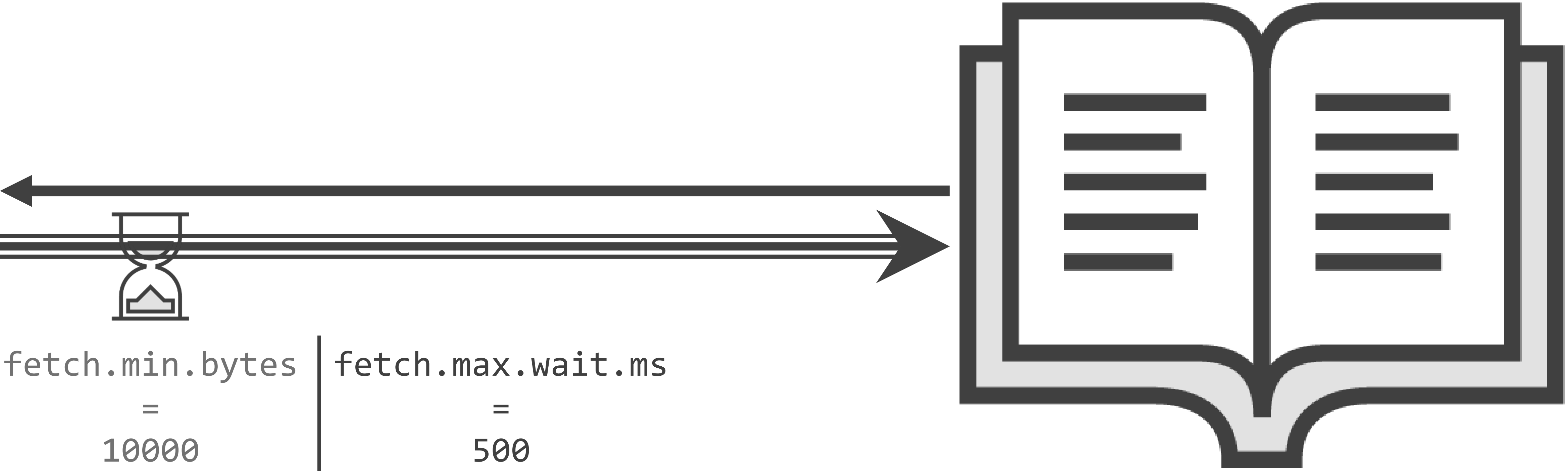



Cluster Load
100
50
0


Cluster

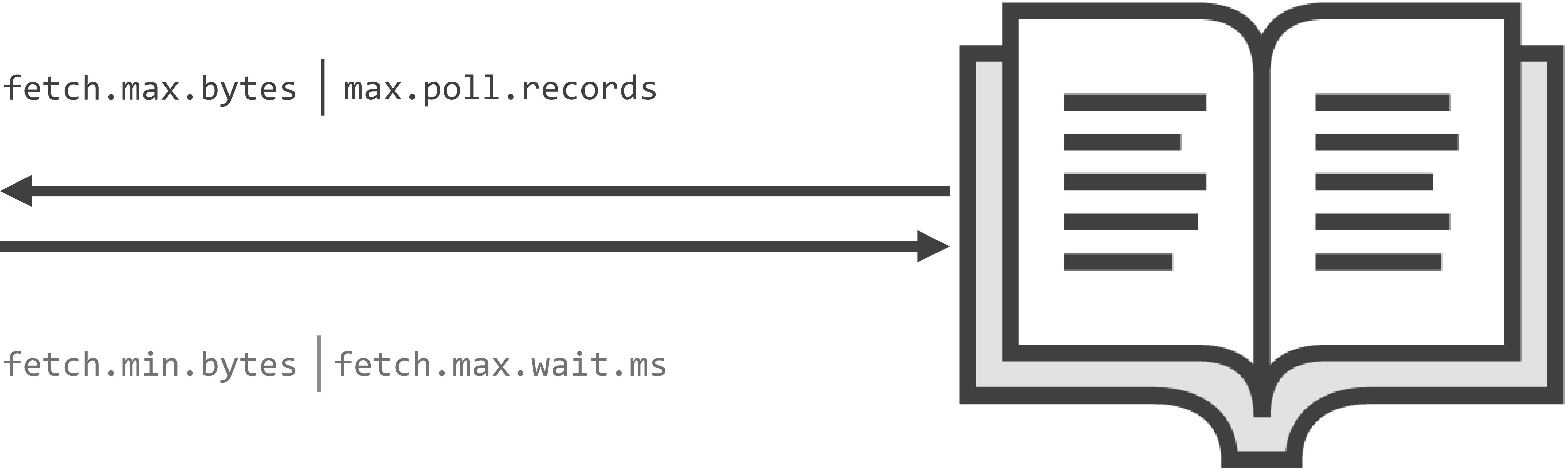


```
fetch.min.bytes
=
10000
```



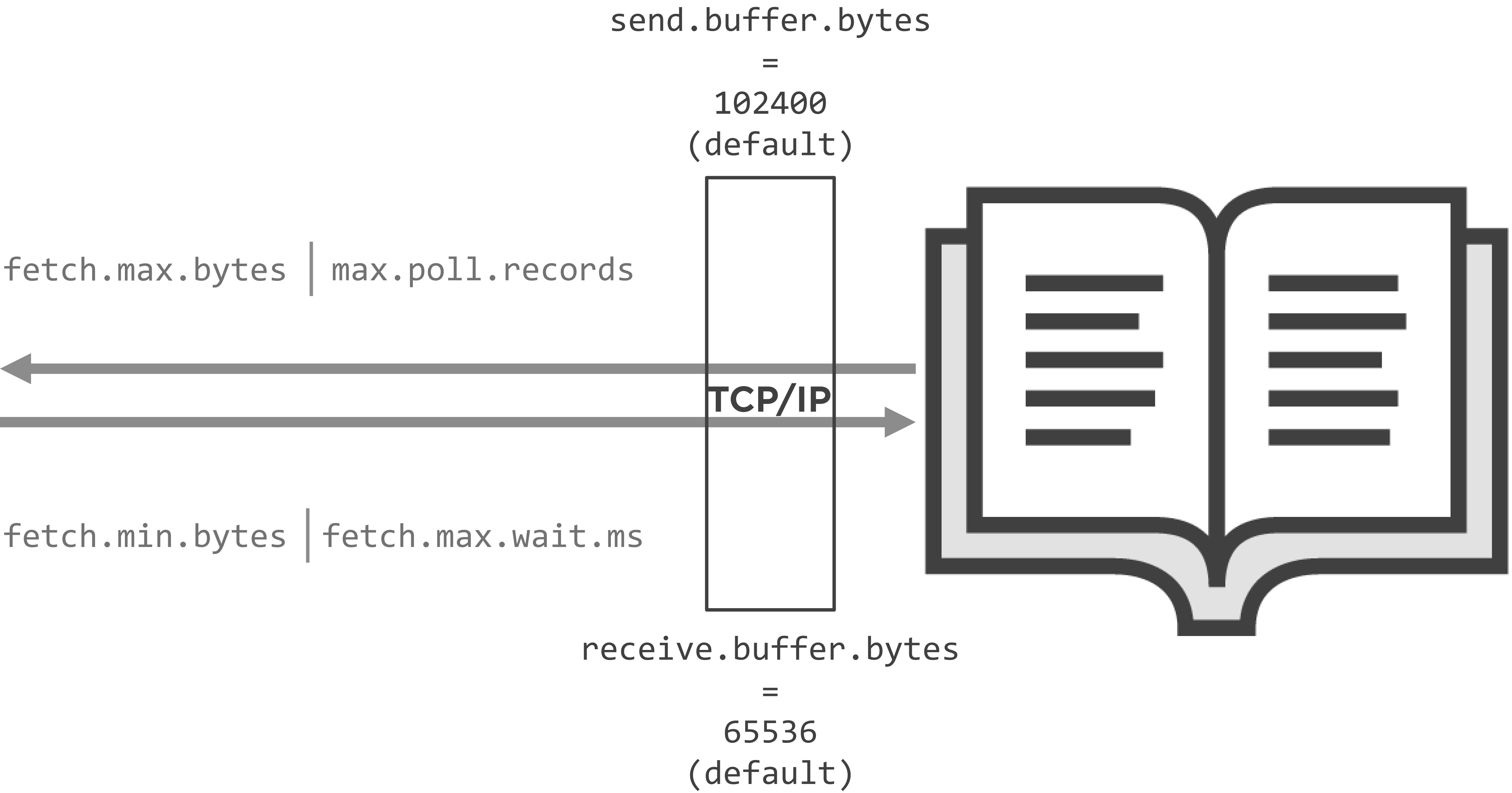
fetch.min.bytes
=
10000

fetch.max.wait.ms
=
500
(default)



fetch.max.bytes | max.poll.records

fetch.min.bytes | fetch.max.wait.ms



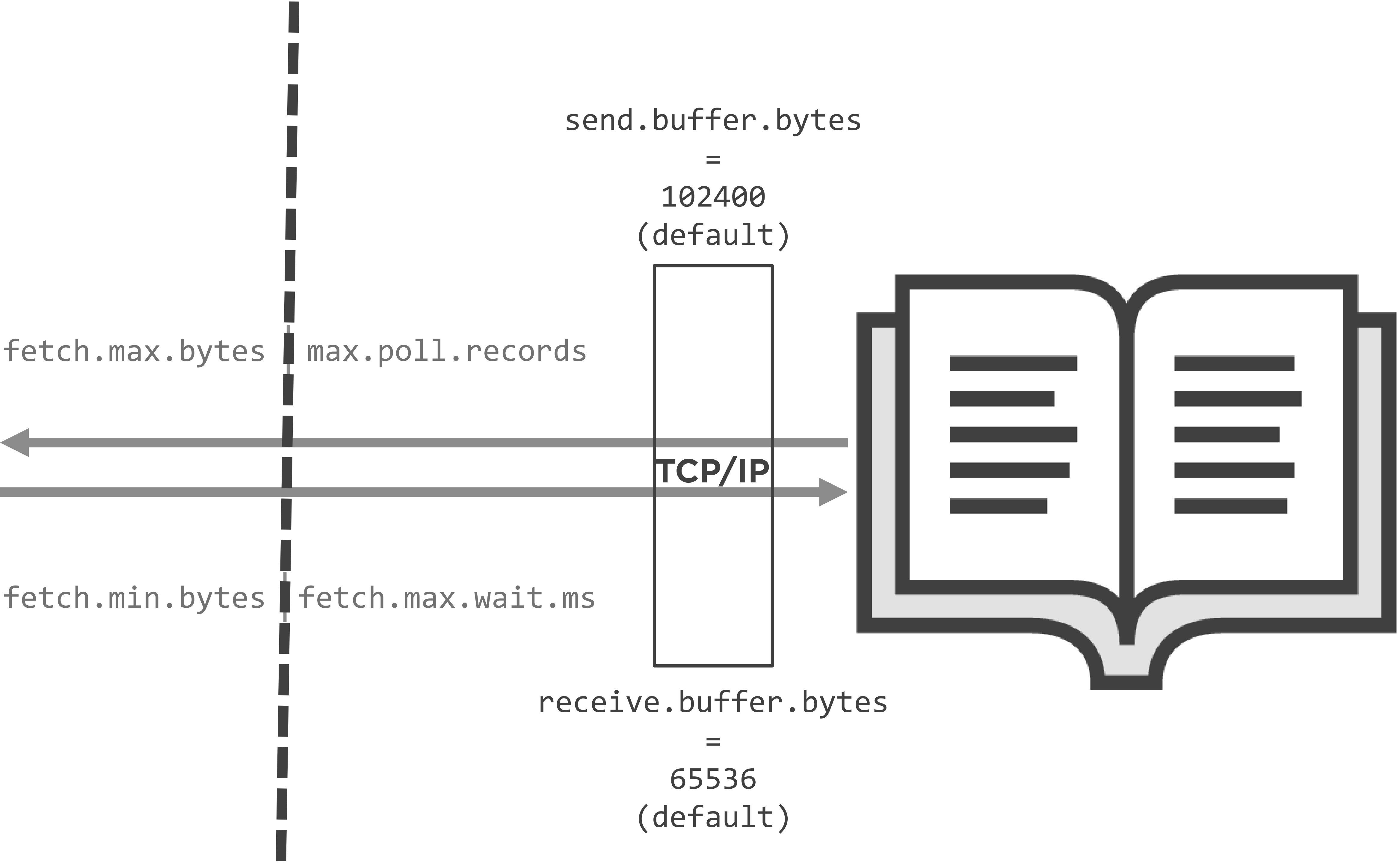
send.buffer.bytes
=
102400
(default)

fetch.max.bytes | max.poll.records

TCP/IP

fetch.min.bytes | fetch.max.wait.ms

receive.buffer.bytes
=
65536
(default)



send.buffer.bytes
=
102400
(default)

fetch.max.bytes

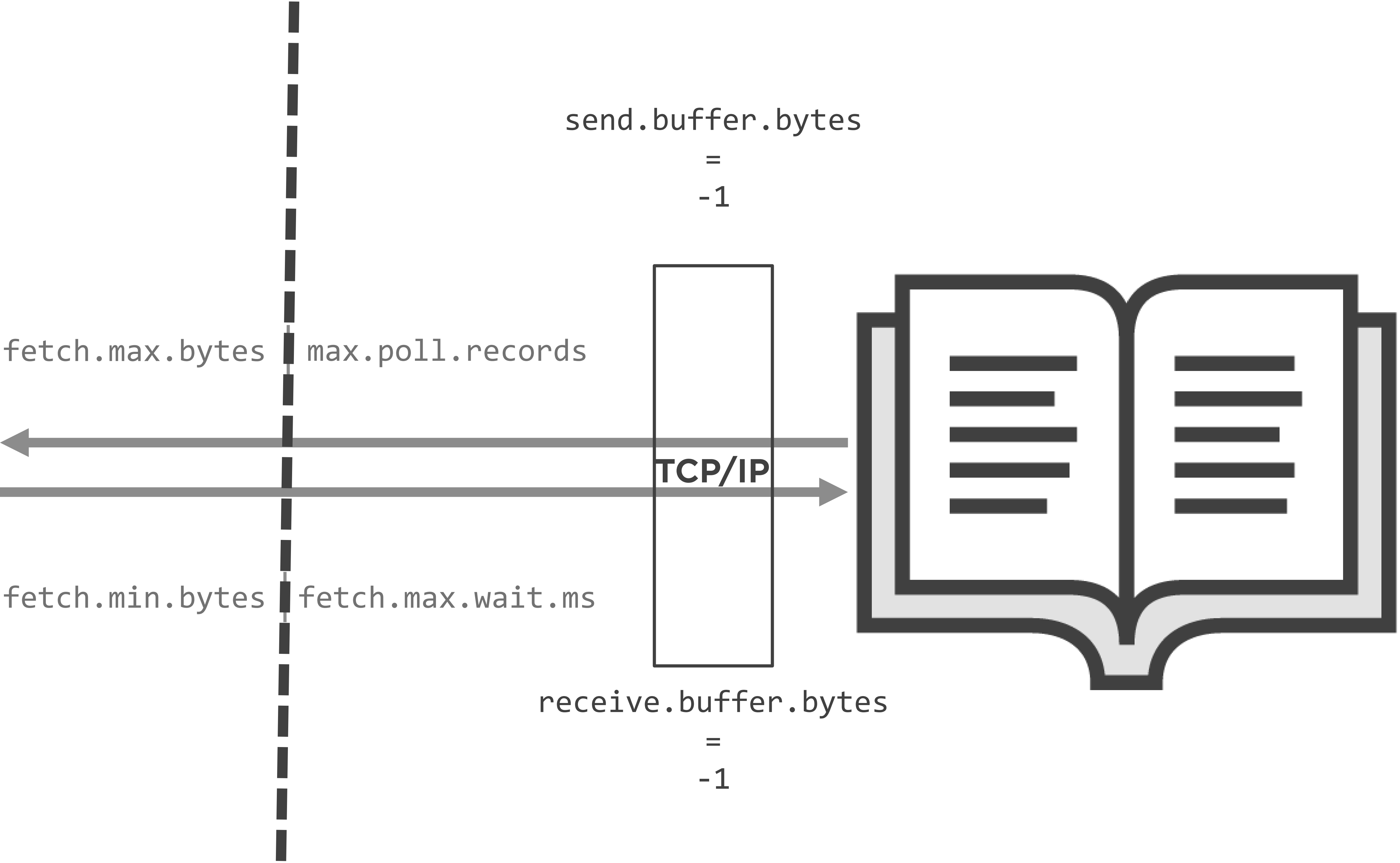
max.poll.records

TCP/IP

fetch.min.bytes

fetch.max.wait.ms

receive.buffer.bytes
=
65536
(default)



send.buffer.bytes
=
-1

fetch.max.bytes

max.poll.records

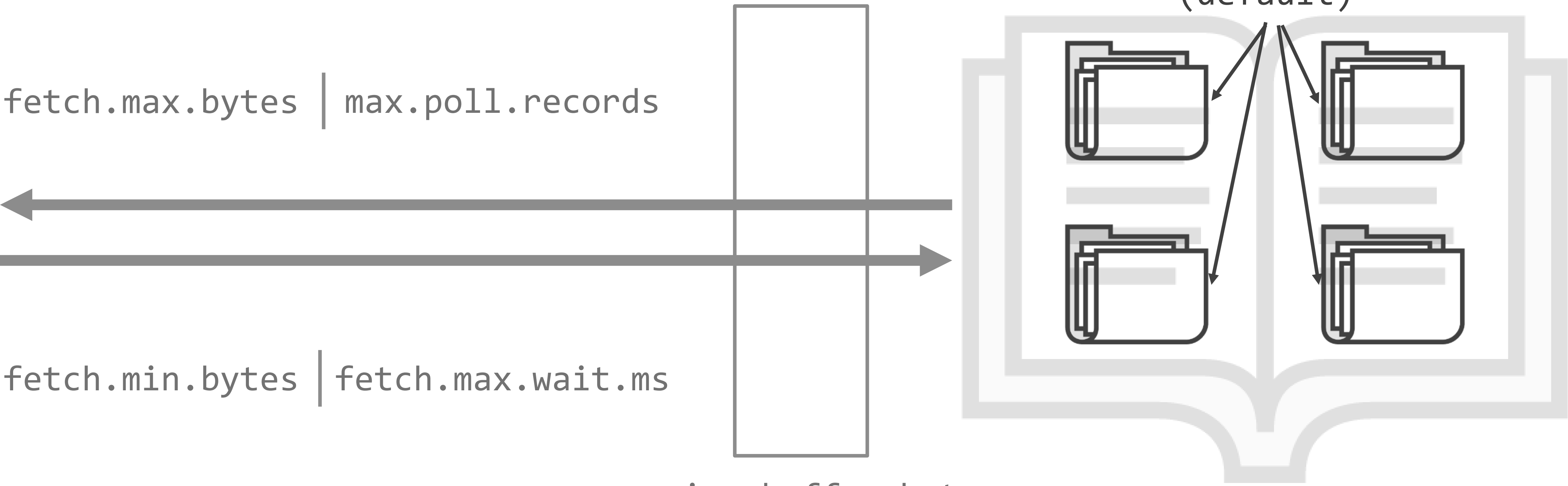
TCP/IP

fetch.min.bytes

fetch.max.wait.ms

receive.buffer.bytes
=
-1

max.message.bytes (topic)
message.max.bytes (broker)
<
max.partition.fetch.bytes
=
1048576
(default)



send.buffer.bytes

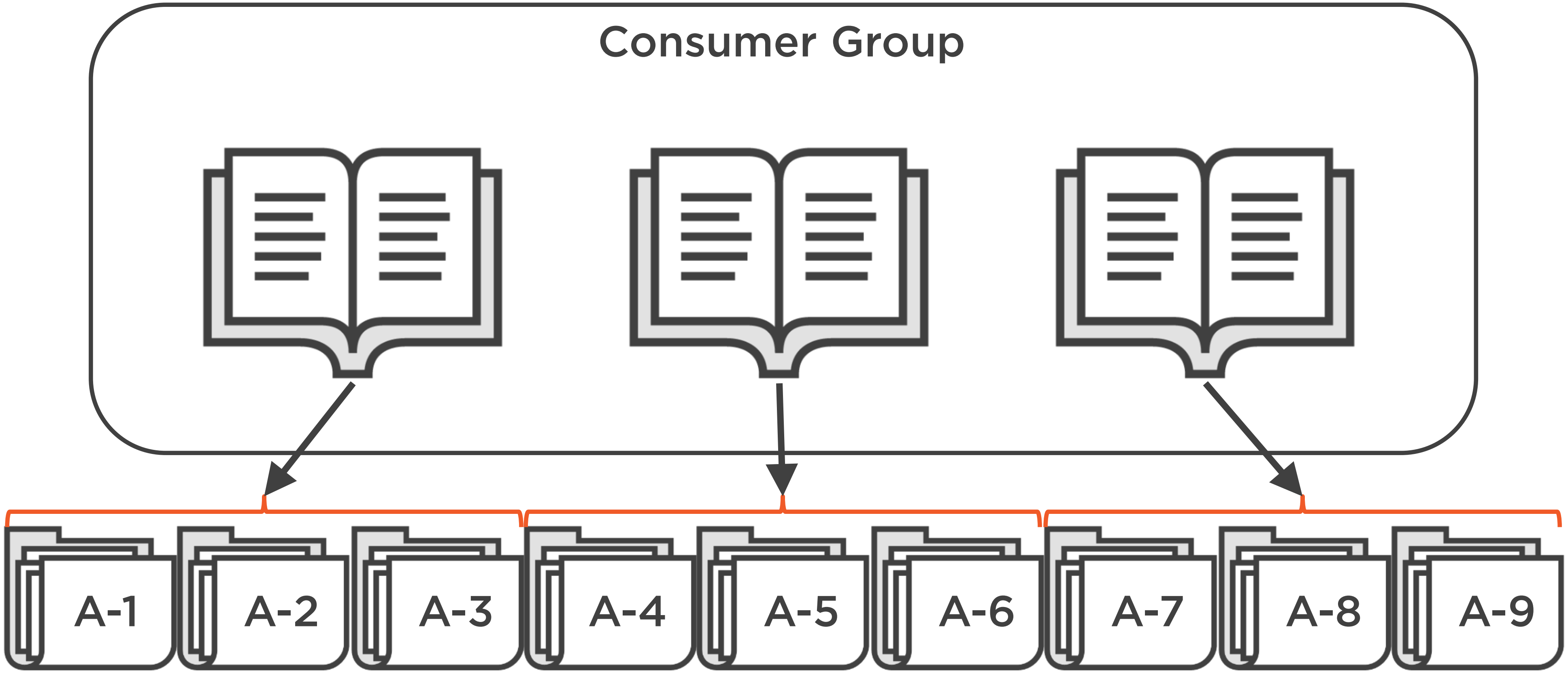
fetch.max.bytes | max.poll.records

fetch.min.bytes | fetch.max.wait.ms

receive.buffer.bytes

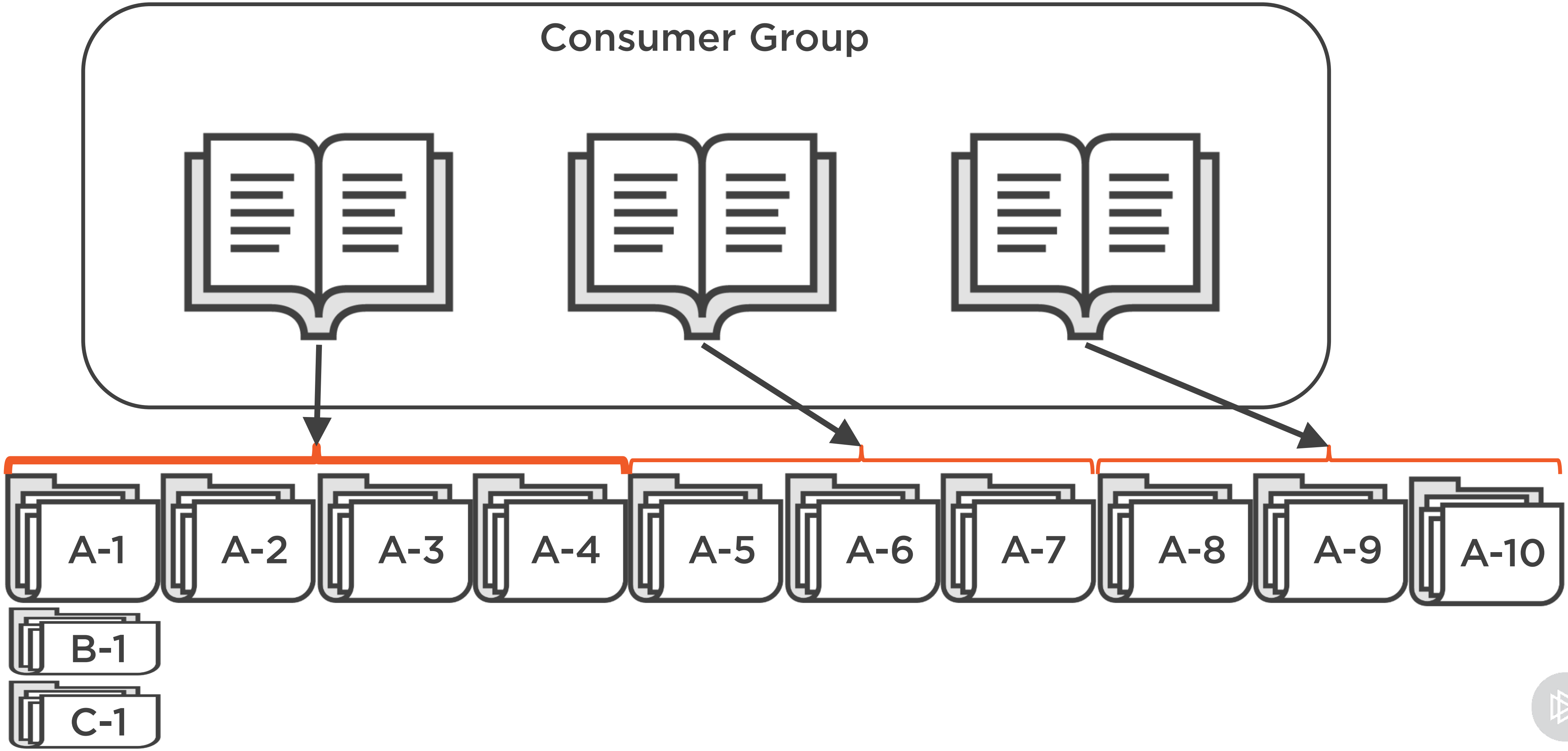
partition.assignment.strategy

Default = `org.apache.kafka.clients.consumer.RangeAssignor`



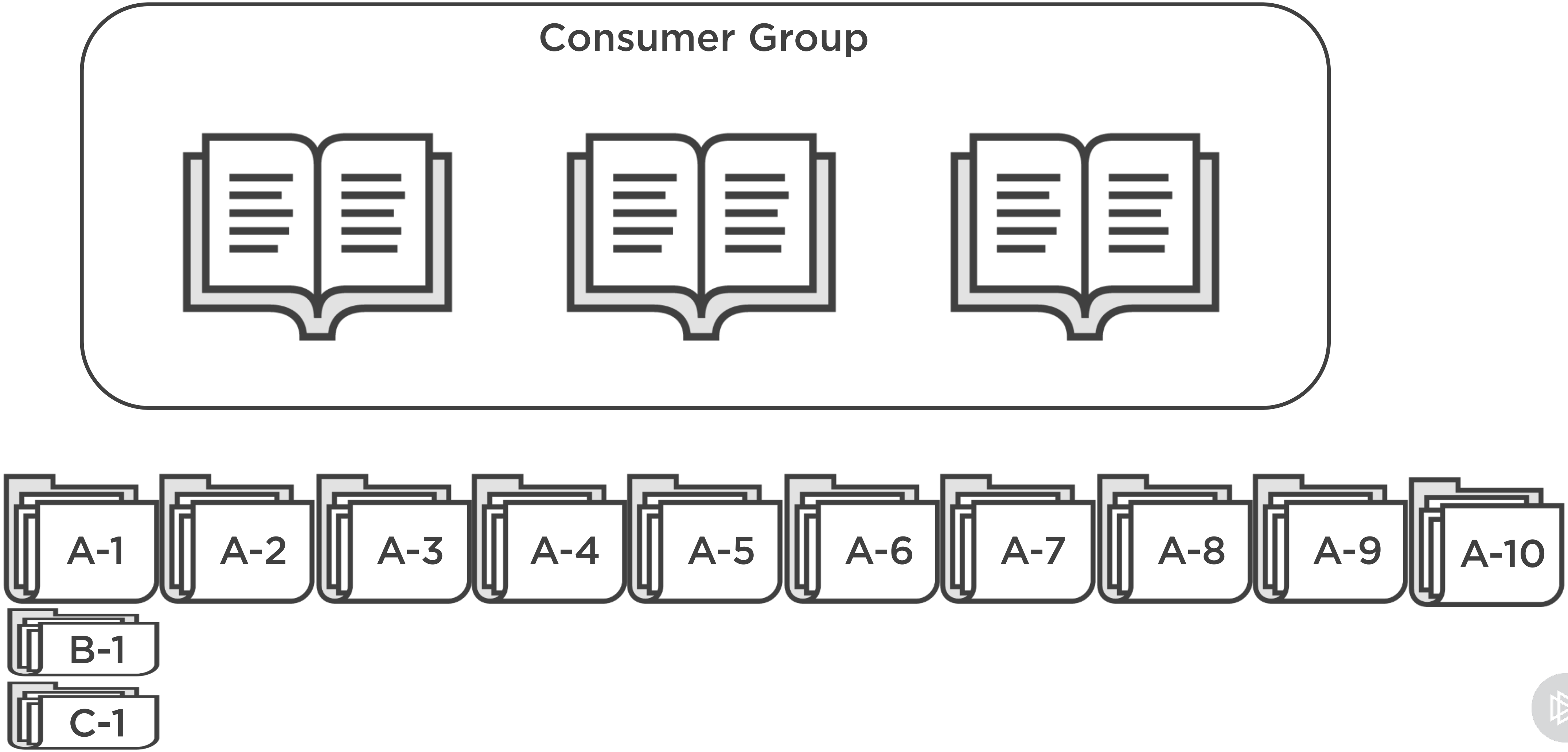
partition.assignment.strategy

Default = `org.apache.kafka.clients.consumer.RangeAssignor`



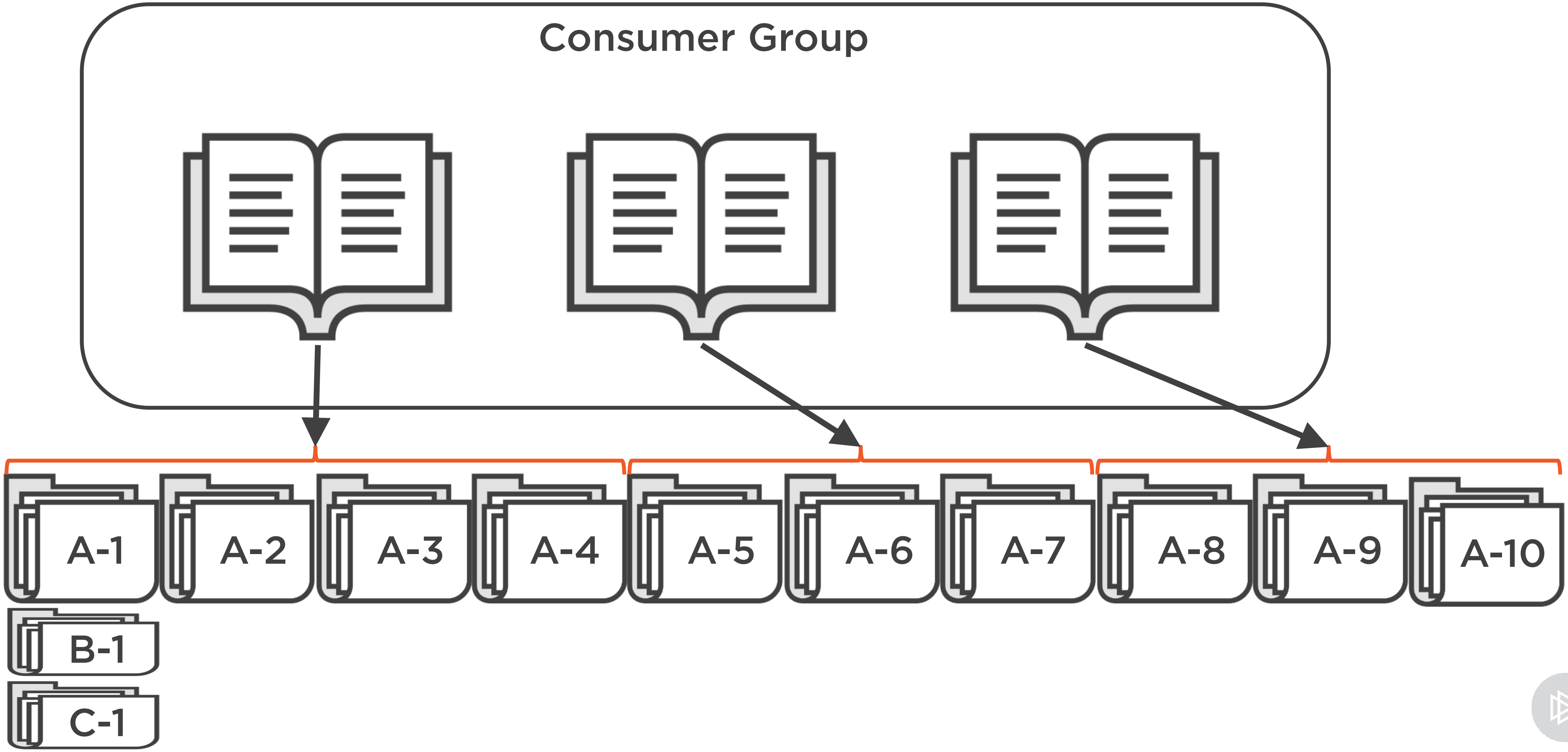
partition.assignment.strategy

`org.apache.kafka.clients.consumer.RoundRobinAssignor`



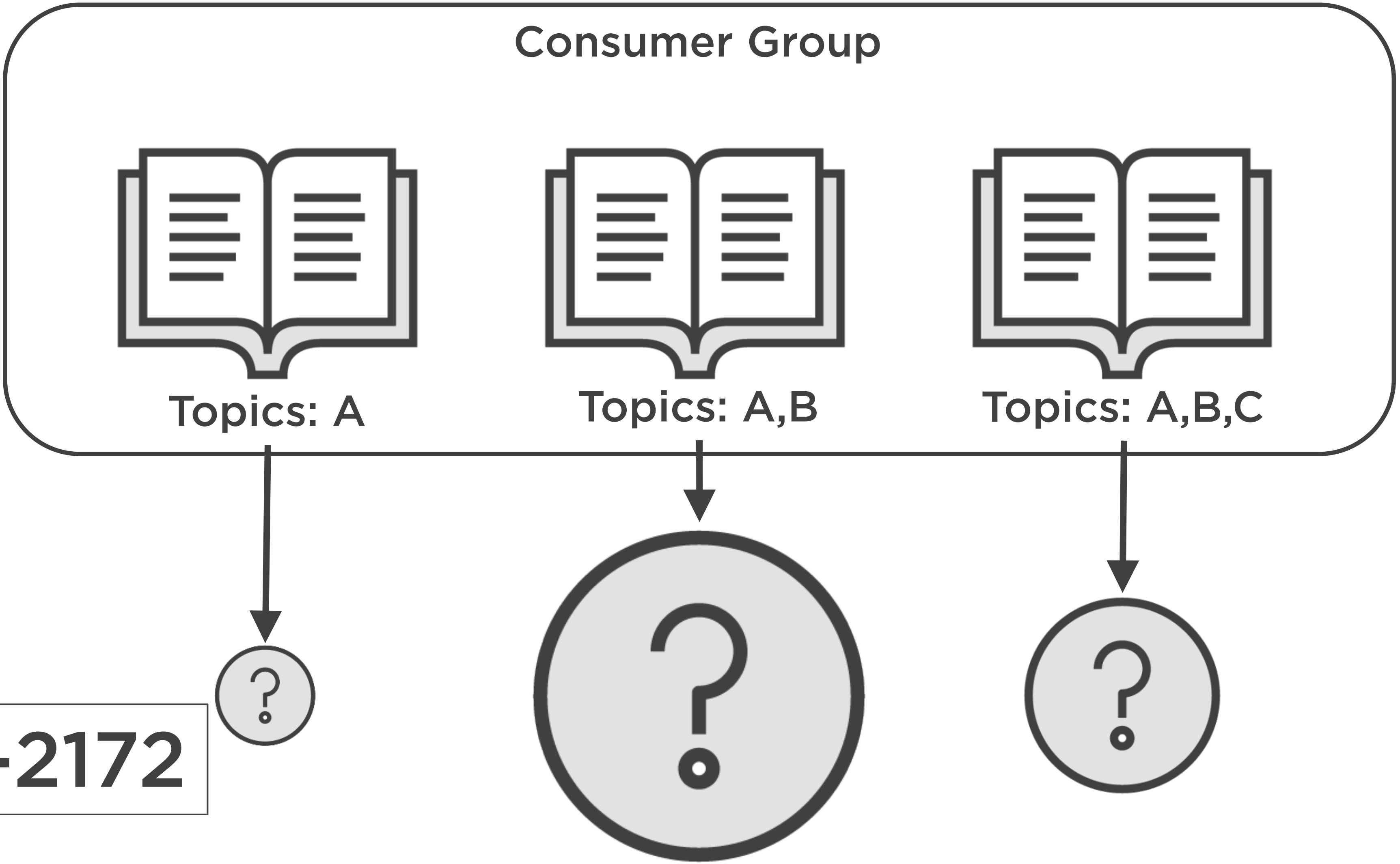
partition.assignment.strategy

`org.apache.kafka.clients.consumer.RoundRobinAssignor`



partition.assignment.strategy

`org.apache.kafka.clients.consumer.RoundRobinAssignor`

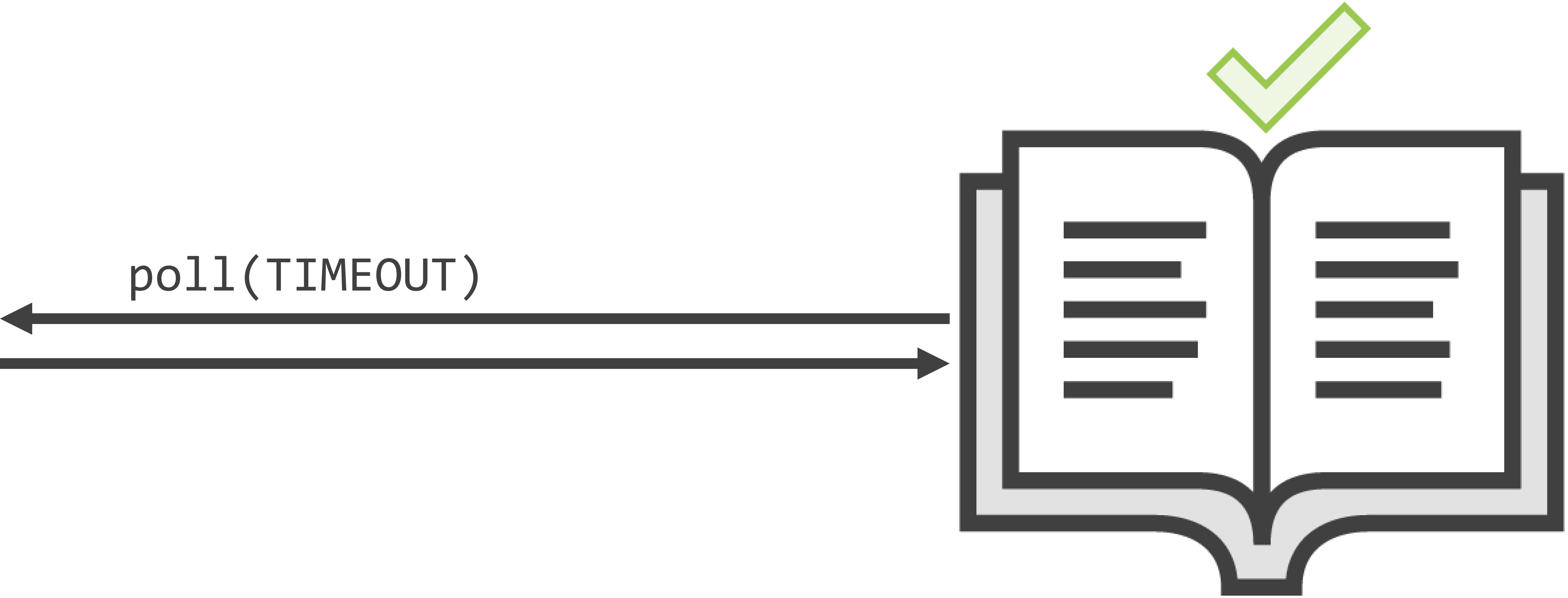


KAFKA-2172

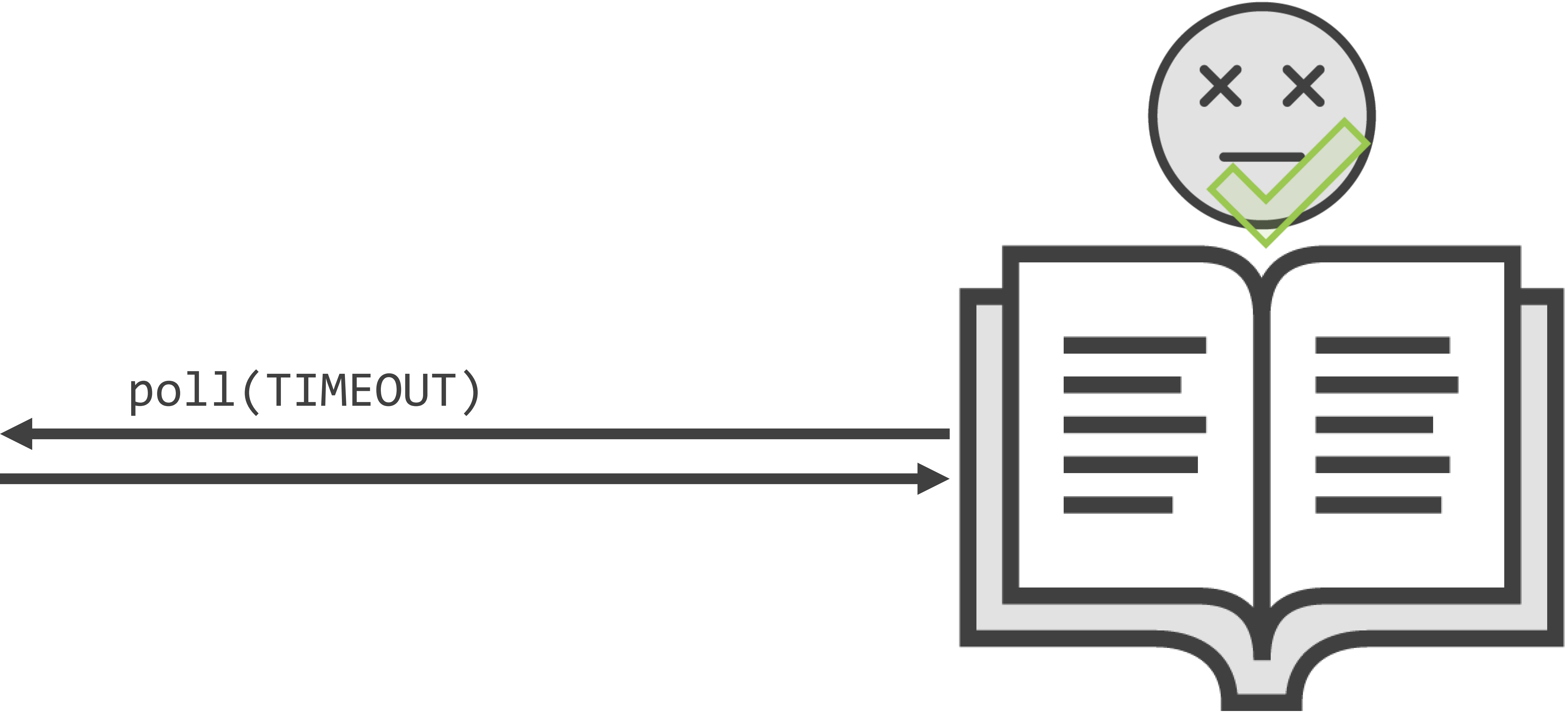
partition.assignment.strategy

```
Coming Soon? -> FairAssignor
```

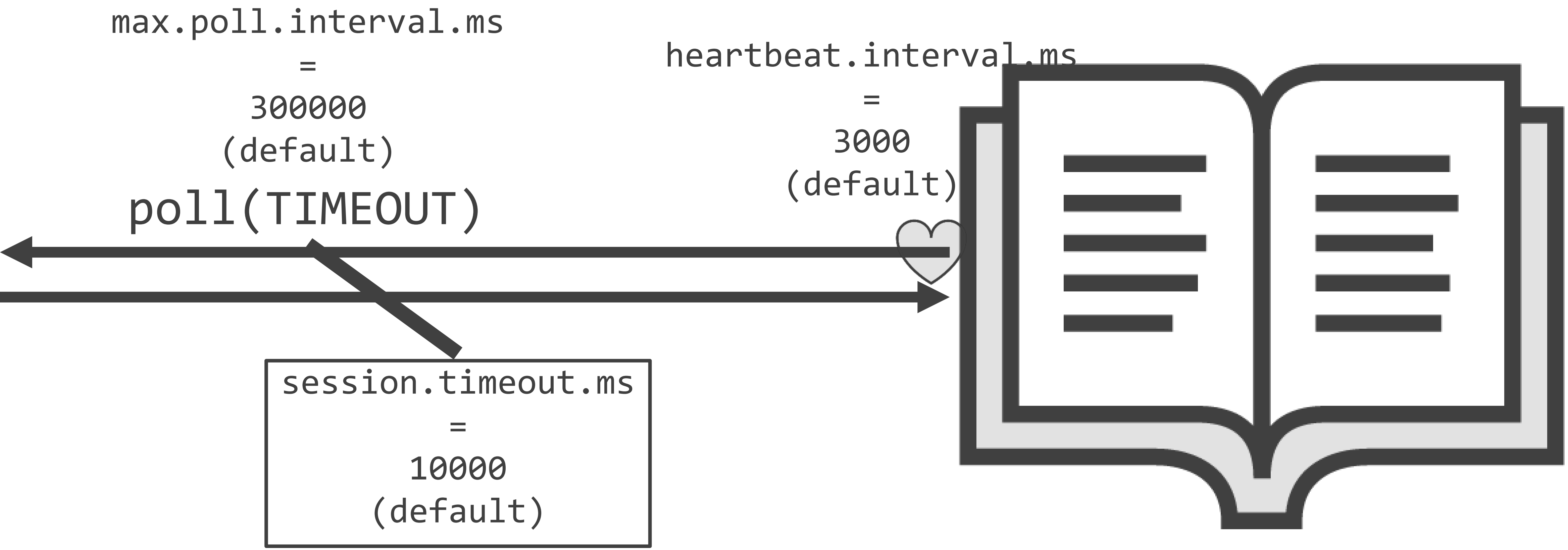
```
org.apache.kafka.clients.consumer.internals.PartitionAssignor
```



poll(TIMEOUT)



poll(TIMEOUT)

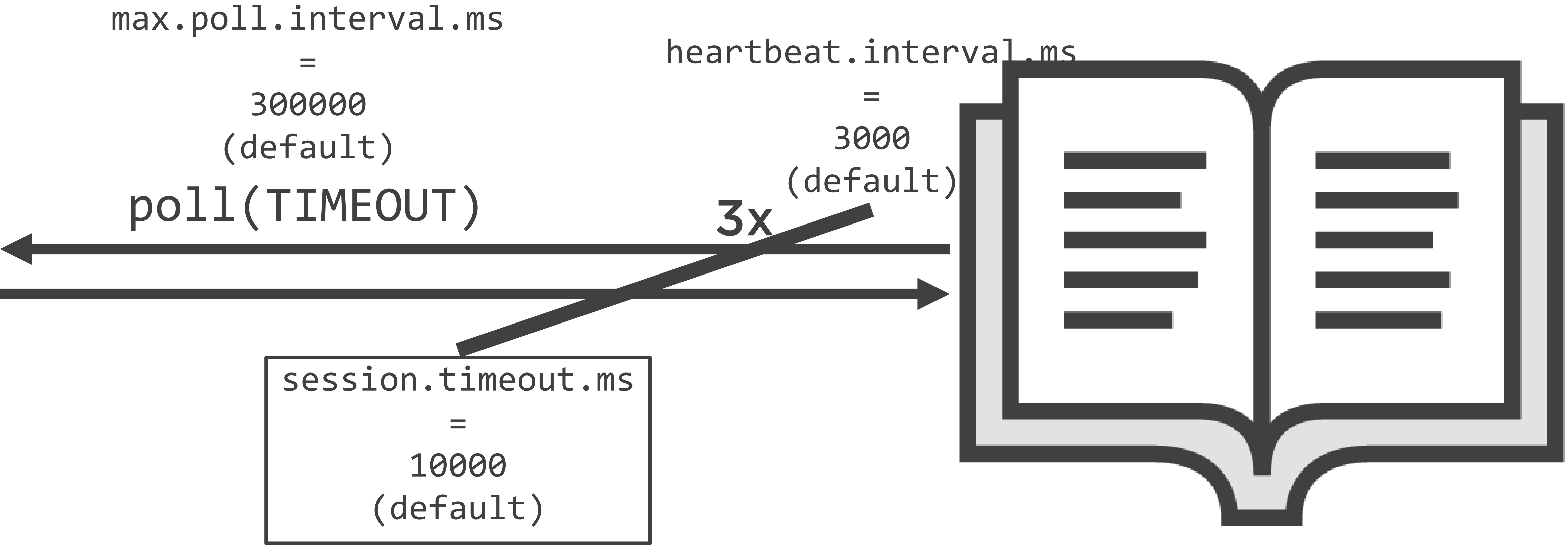


max.poll.interval.ms
=
300000
(default)

poll(TIMEOUT)

heartbeat.interval.ms
=
3000
(default)

session.timeout.ms
=
10000
(default)



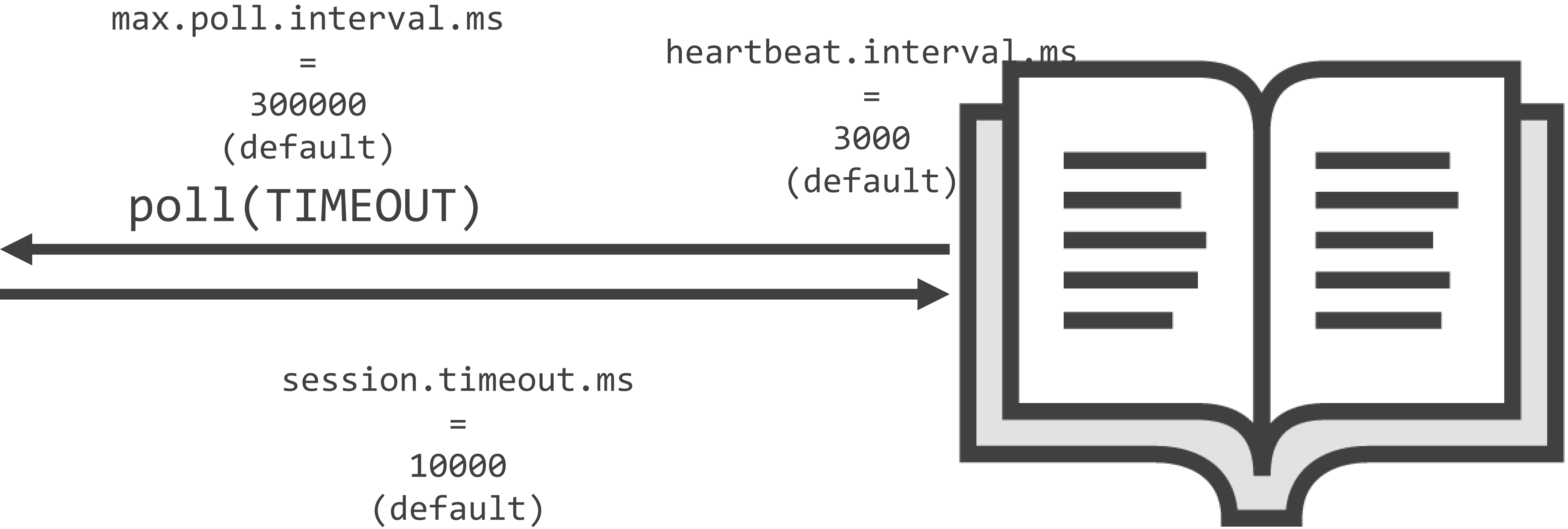
max.poll.interval.ms
=
300000
(default)

poll(TIMEOUT)

heartbeat.interval.ms
=
3000
(default)

3x

session.timeout.ms
=
10000
(default)

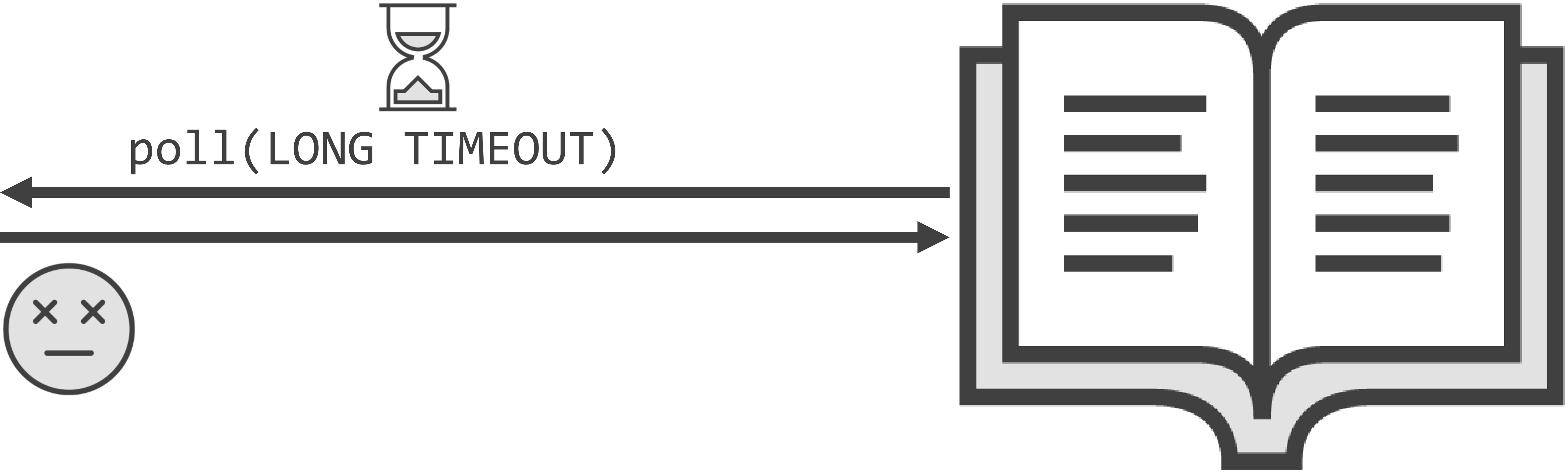


max.poll.interval.ms
=
300000
(default)

poll(TIMEOUT)

heartbeat.interval.ms
=
3000
(default)

session.timeout.ms
=
10000
(default)



poll(LONG TIMEOUT)

default.api.timeout.ms

```
consumer.commitSync()
         commitSync(offsets)
         position(partition)
         committed(partition)
```

default.api.timeout.ms

```
consumer.commitSync()
         commitSync(offsets)
         position(partition)
         committed(partition)
```

KIP-266

```
default.api.timeout.ms
=
60000
(default)
```

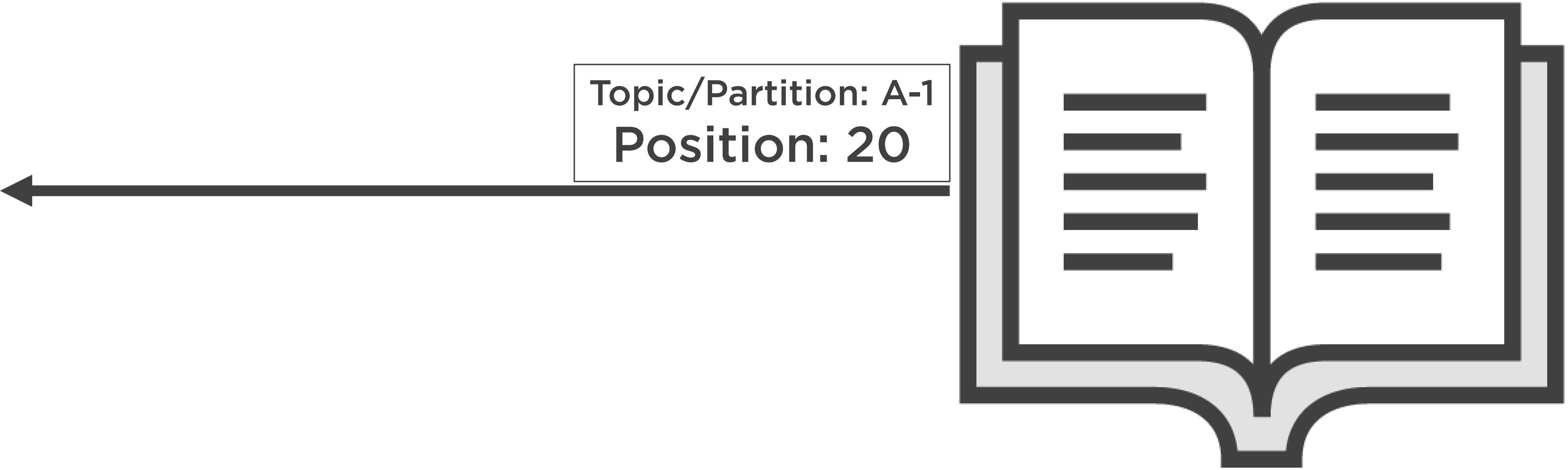
```
consumer.commitSync()........................ ⚠
         commitSync(offsets)
         position(partition)
         committed(partition)
```

Configurations, Configurations, Configurations

- `[ssl/sasl]……`
- `client.dns.lookup`
- `request.timeout.ms`
- …

Consumer Offset Tracking

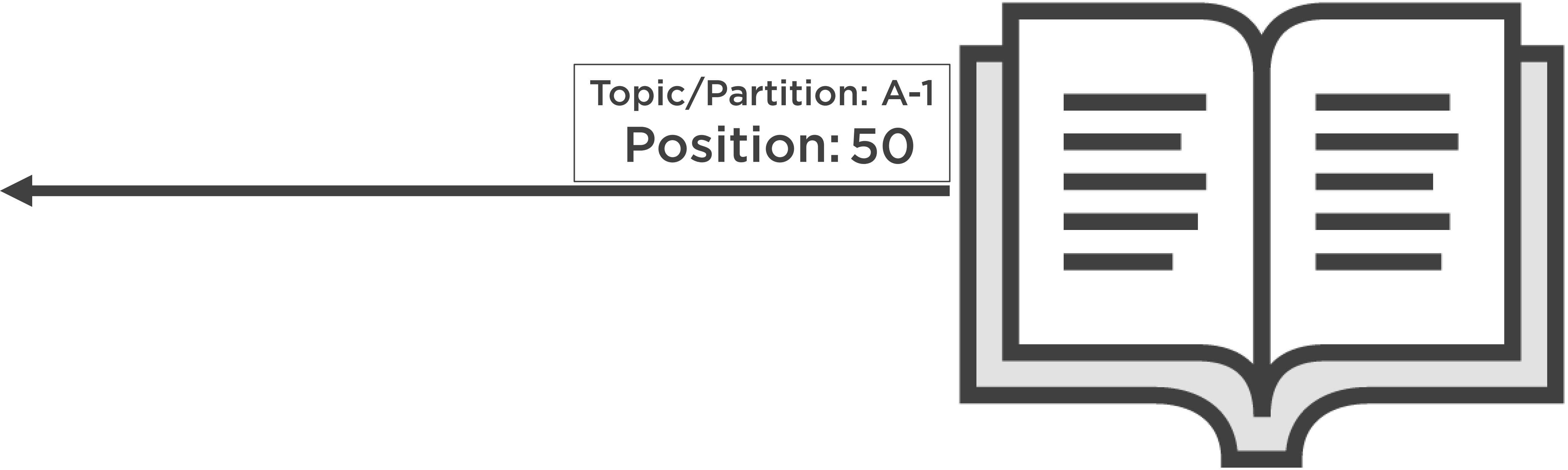
Offset Tracking



Topic/Partition: A-1

Position: 20

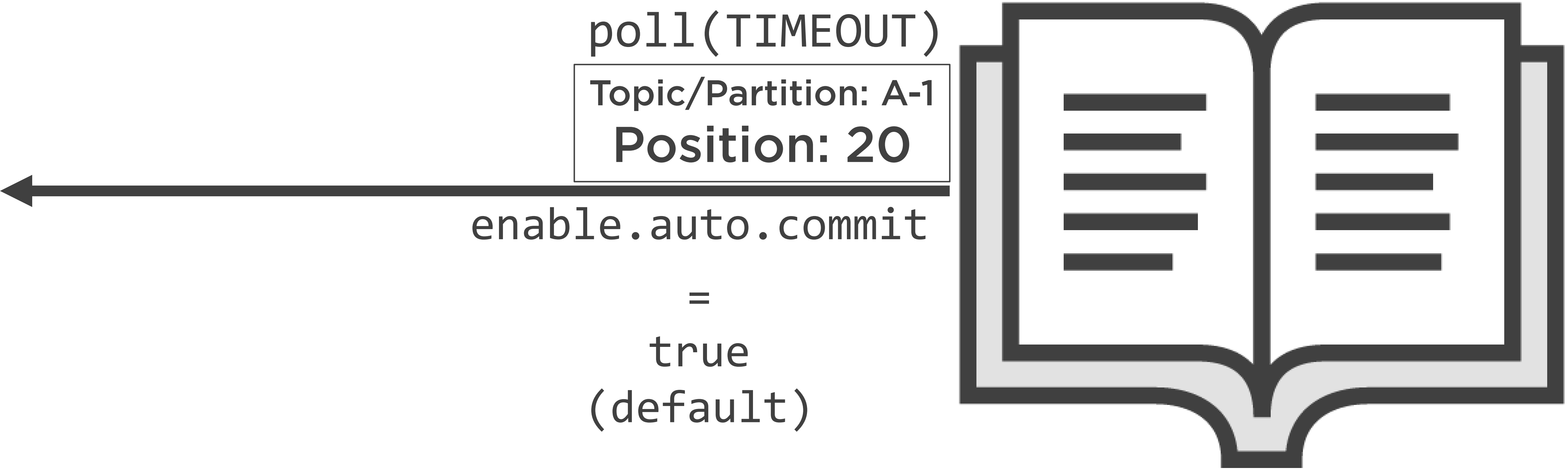
Offset Tracking



Topic/Partition: A-1

Position: 50

Offset Tracking



poll(TIMEOUT)

Topic/Partition: A-1

Position: 20

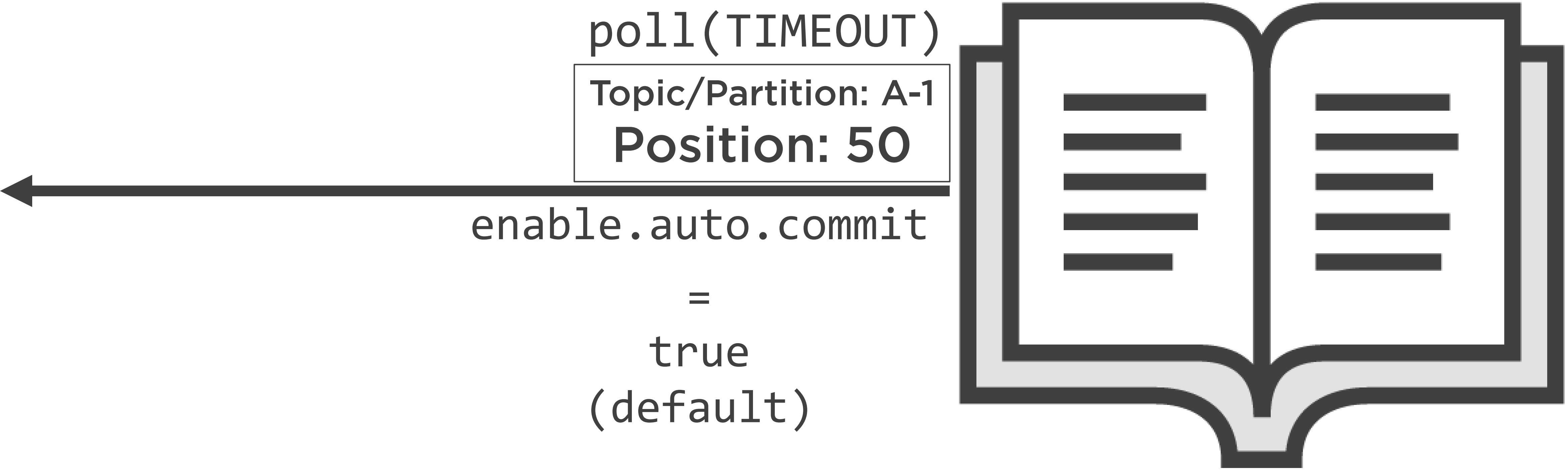
enable.auto.commit

=

true

(default)

Offset Tracking



`poll(TIMEOUT)`

Topic/Partition: A-1

Position: 50

`enable.auto.commit`

`=`

`true`

`(default)`

Offset Tracking

```
enable.auto.commit
=
true
(default)
```

```
auto.commit.interval.ms
=
5000
(default)
```

Offset Tracking

Overall Time: 4 sec
Min Next Commit: 5 sec

```
poll(4 sec)
```

```
enable.auto.commit
=
true
(default)
```

```
auto.commit.interval.ms
=
5000
(default)
```

Offset Tracking

Overall Time: 4 sec
Min Next Commit: 5 sec

poll(4 sec)

Topic/Partition: A-1
Position: 20

enable.auto.commit

=

true

(default)

---

auto.commit.interval.ms

=

5000

(default)

Offset Tracking

Overall Time: 8 sec
Min Next Commit: 5 sec

```
enable.auto.commit
=
true
(default)
```

```
auto.commit.interval.ms
=
5000
(default)
```

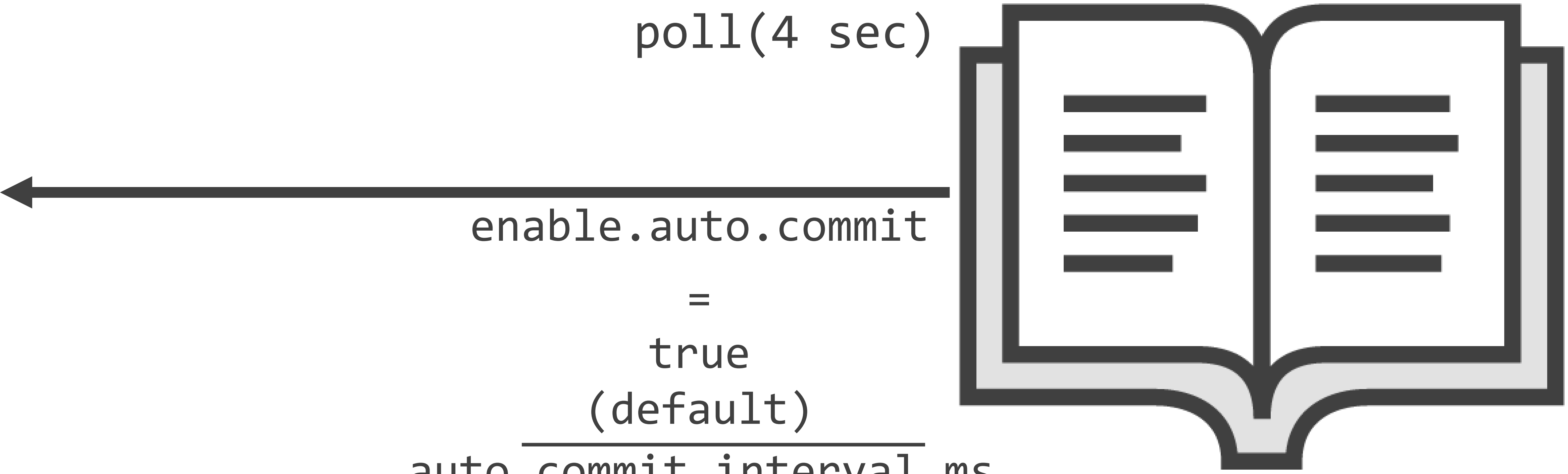
Offset Tracking

Overall Time: 8 sec
Min Next Commit: 13 sec

```
enable.auto.commit
        =
       true
    (default)
-----------------------
auto.commit.interval.ms
        =
       5000
    (default)
```

Offset Tracking

Overall Time: 8 sec
Min Next Commit: 13 sec



```
poll(4 sec)

enable.auto.commit
=
true
(default)
```

```
auto.commit.interval.ms
=
5000
(default)
```

Offset Tracking

Overall Time: 12 sec
Min Next Commit: 13 sec

```
enable.auto.commit
=
true
(default)
```

```
auto.commit.interval.ms
=
5000
(default)
```

Offset Tracking

Overall Time: 12 sec
Min Next Commit: 13 sec

```
poll(4 sec)
```

Topic/Partition: A-1
Position: 50

```
enable.auto.commit
=
true
(default)
```

```
auto.commit.interval.ms
=
5000
(default)
```

Offset Tracking

Overall Time: 16 sec
Min Next Commit: 13 sec

```
enable.auto.commit
        =
      true
   (default)
auto.commit.interval.ms
        =
      5000
   (default)
```

Offset Tracking

`close()`

Topic/Partition: A-1

Position: 100

`enable.auto.commit`

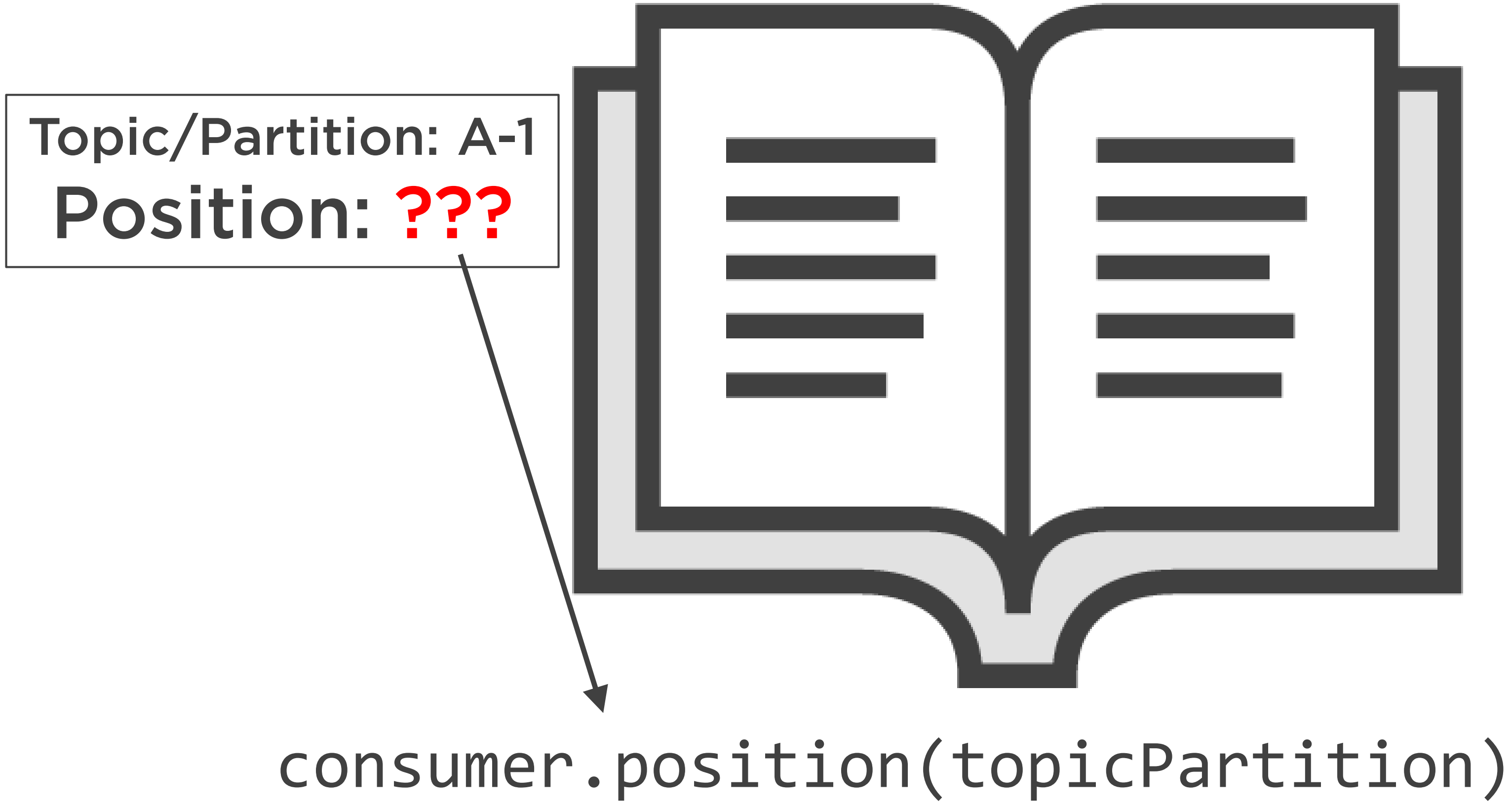
`=`

`true`

`(default)`

Consumer Offset Retrieval

auto.offset.reset



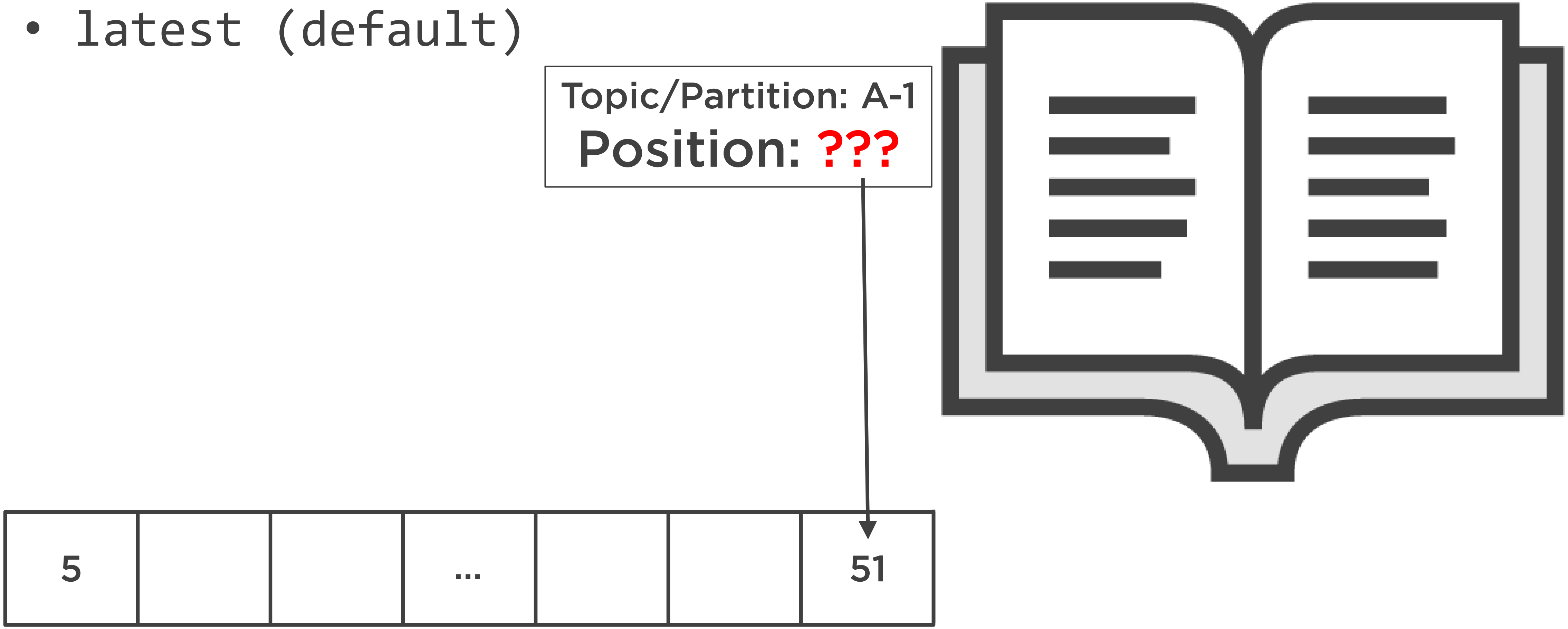
Topic/Partition: A-1

Position: ???

`consumer.position(topicPartition)`

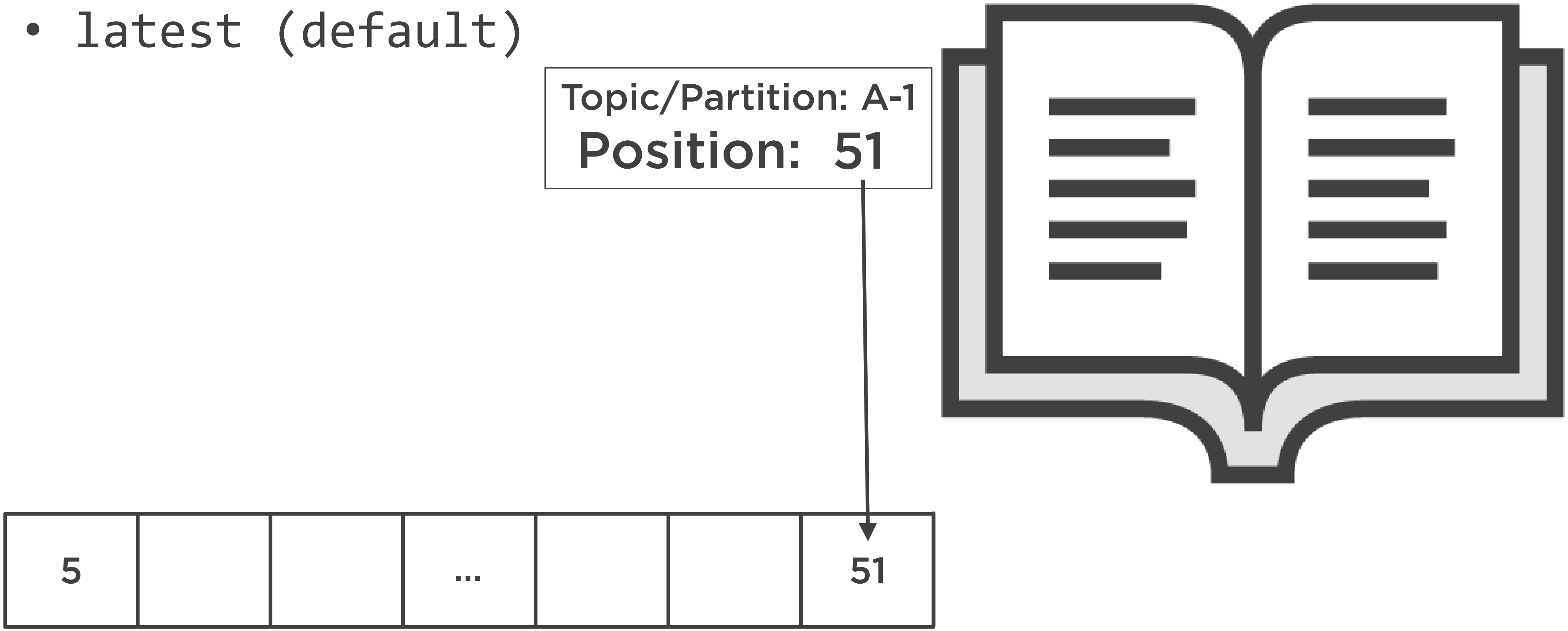
auto.offset.reset

- latest (default)



auto.offset.reset

- latest (default)



Topic/Partition: A-1

Position: 51

5			...			51

auto.offset.reset

- latest (default)

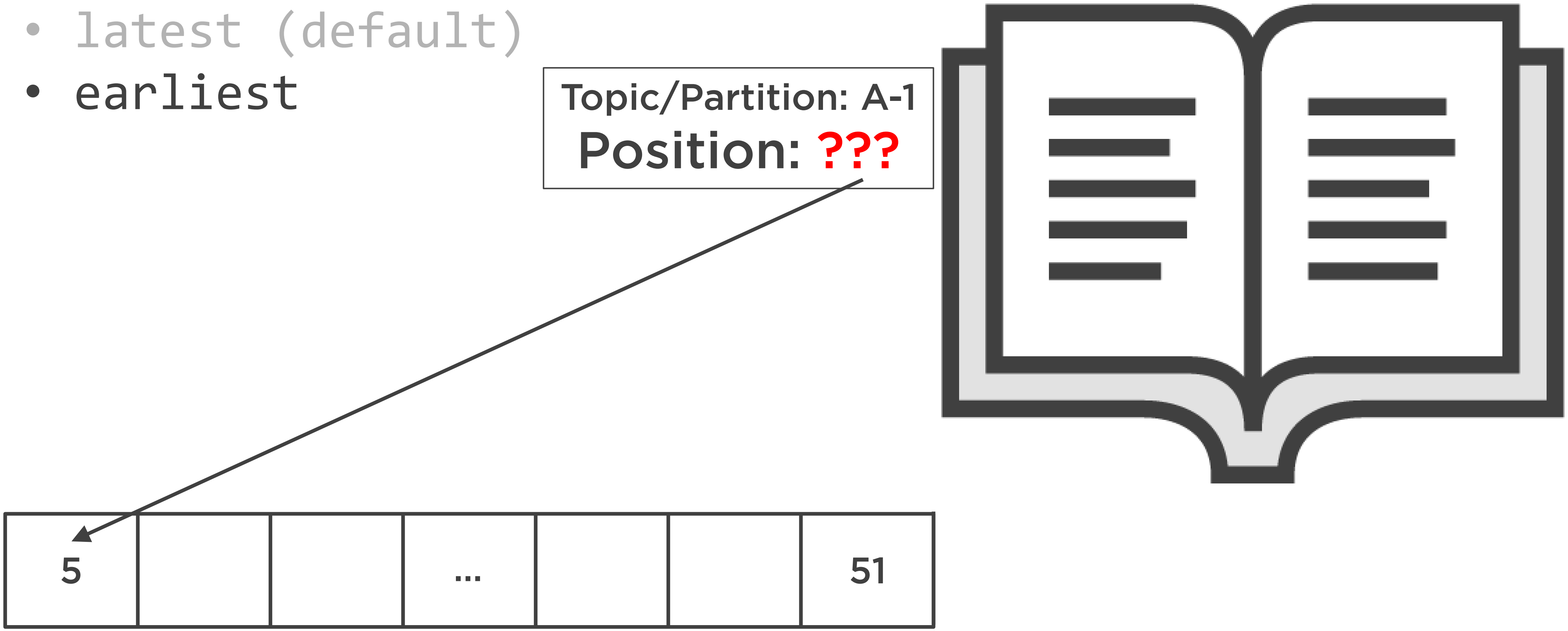
Topic/Partition: A-1

Position: ???

5			...			51

auto.offset.reset

- latest (default)
- earliest



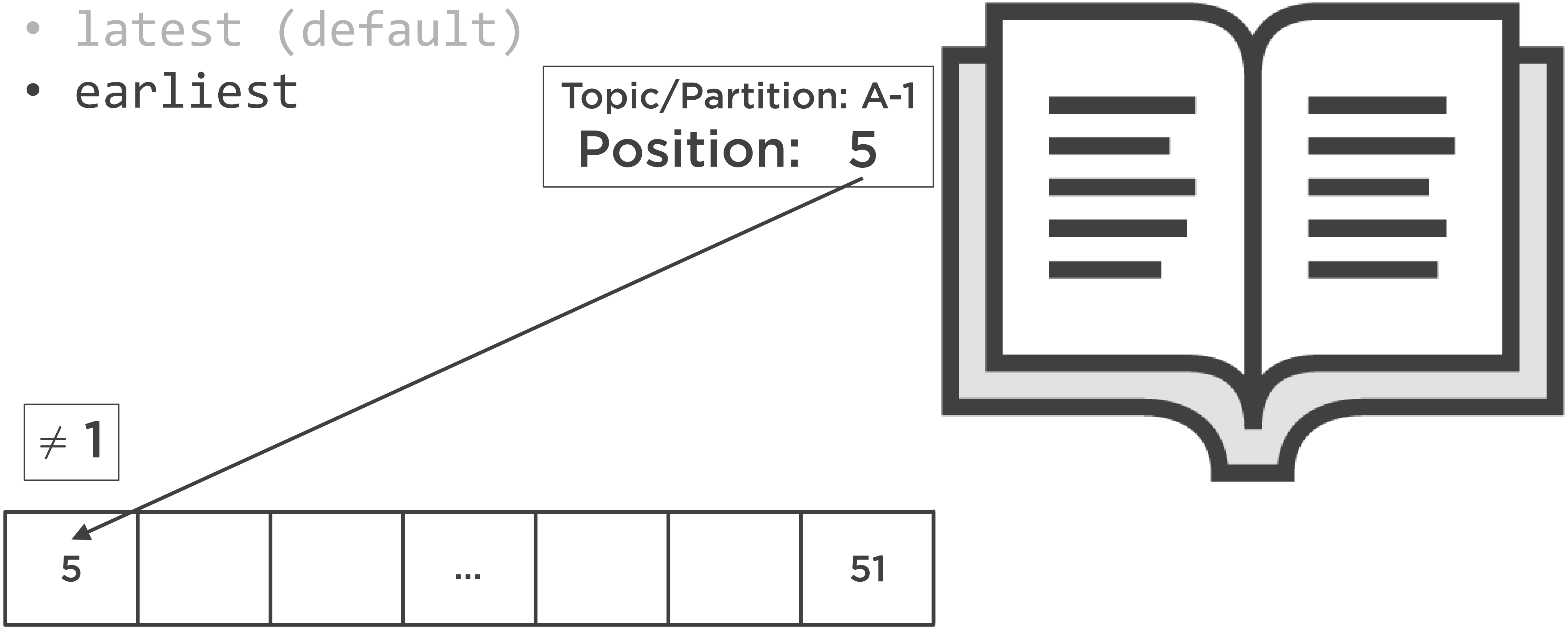
Topic/Partition: A-1

Position: ???

5			...			51

auto.offset.reset

- latest (default)
- earliest



Topic/Partition: A-1
Position: 5

≠ 1

5

...

51

auto.offset.reset

- latest (default)
- earliest

Topic/Partition: A-1

Position: ???

5			...			51

auto.offset.reset

- `latest (default)`
- `earliest`
- none

Topic/Partition: A-1

Position: ???

5			...			51

auto.offset.reset

- latest (default)
- earliest
- none

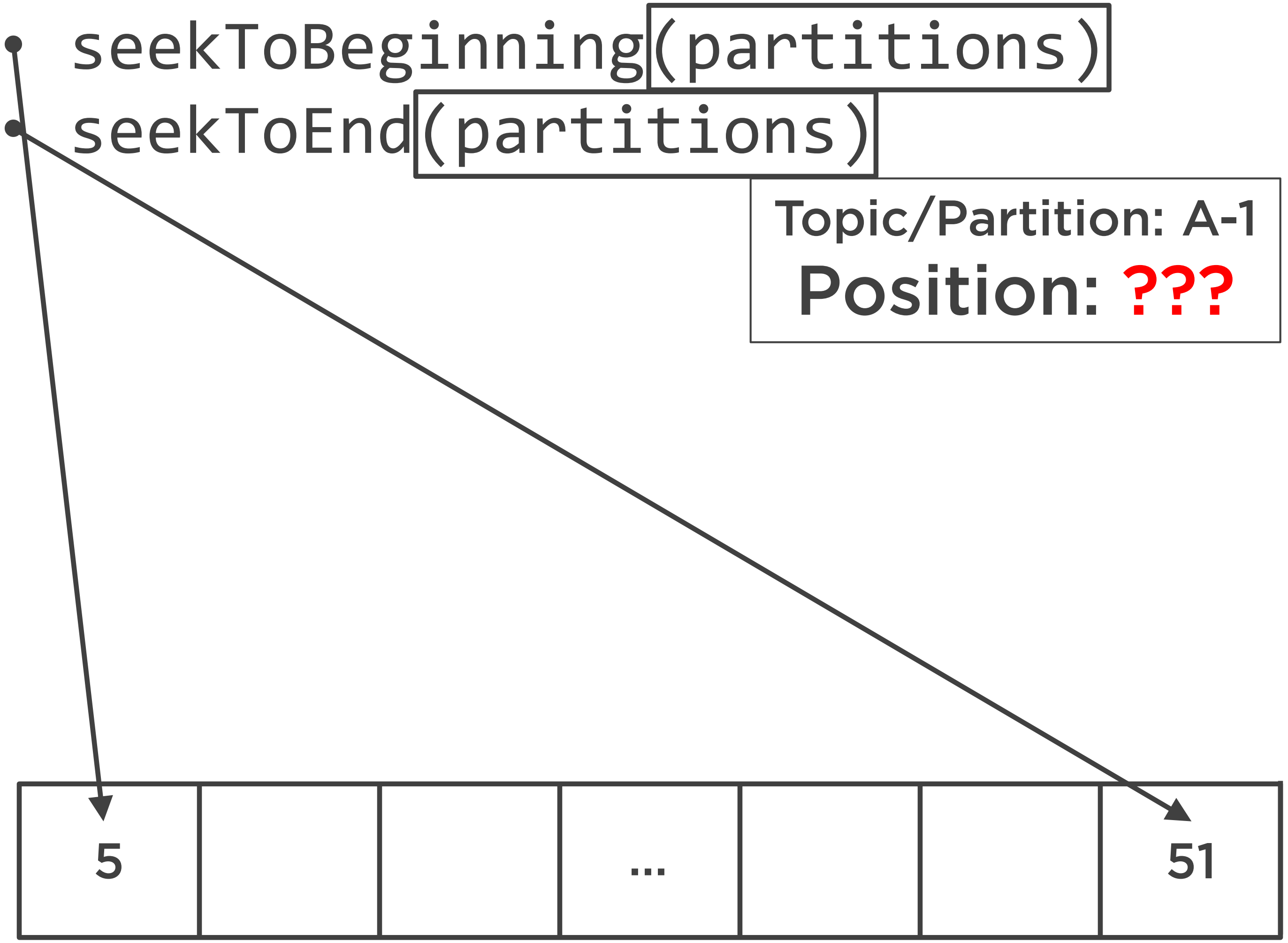
Topic/Partition: A-1

Position: ⚠

5			...			51

Offset Seeking

- `seekToBeginning(partitions)`
- `seekToEnd(partitions)`



Topic/Partition: A-1

Position: ???

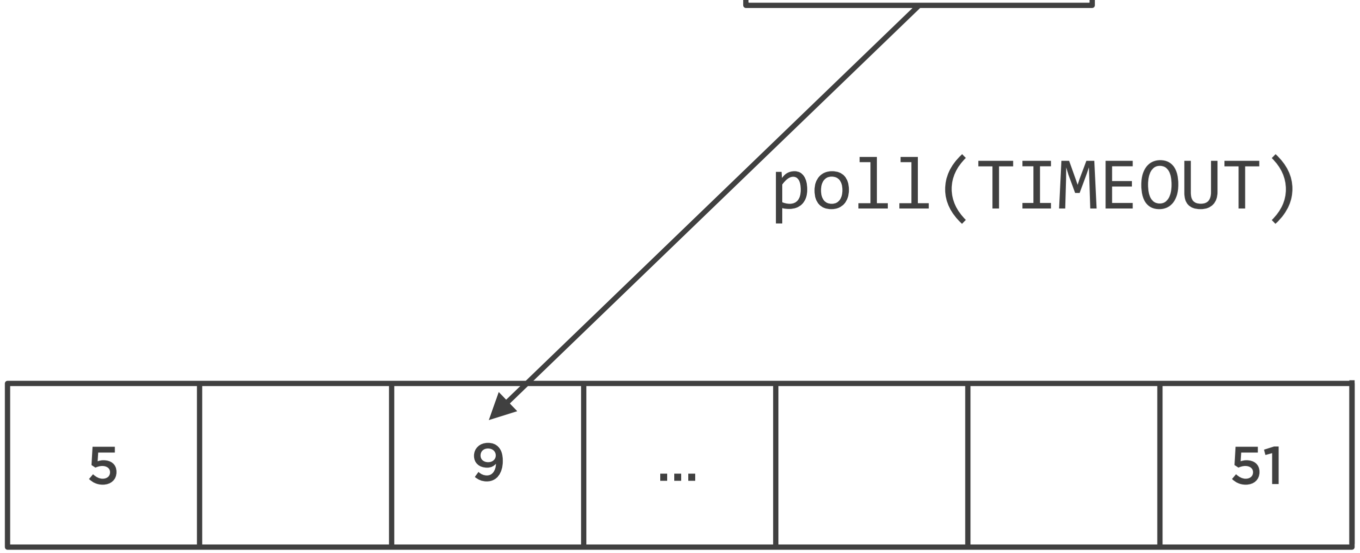
Offset Seeking

- `seekToBeginning(partitions)`
- `seekToEnd(partitions)`

Topic/Partition: A-1
Position: ???

- `seek(partitions, position)`

`poll(TIMEOUT)`



5		9	…			51

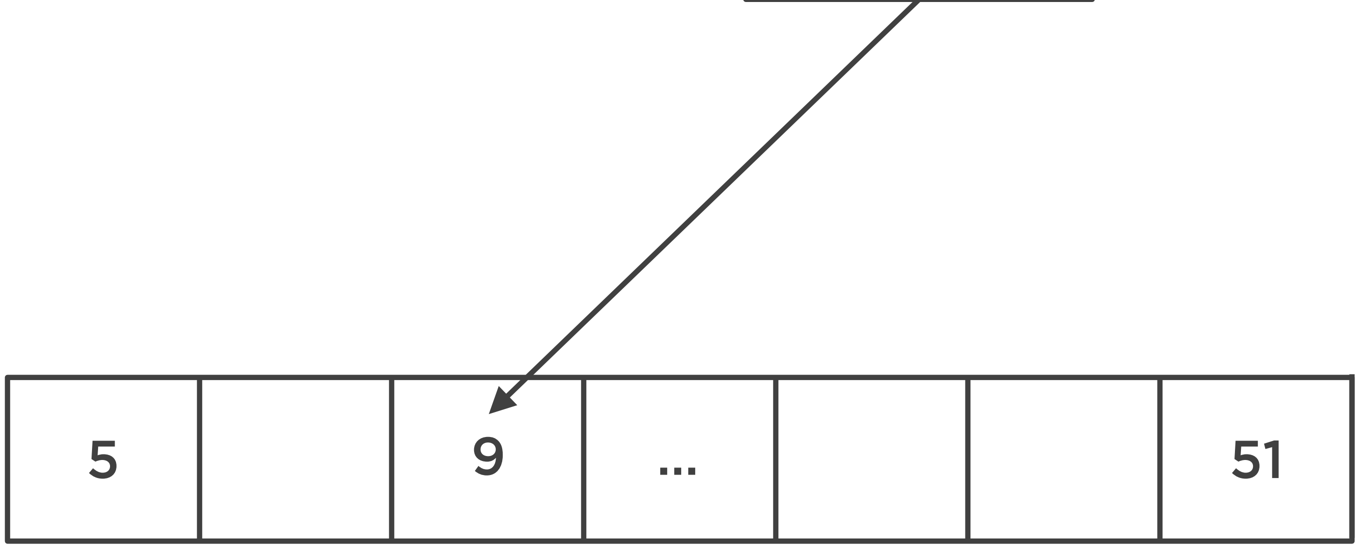
Offset Seeking

- seekToBeginning(partitions)
- seekToEnd(partitions)

Topic/Partition: A-1

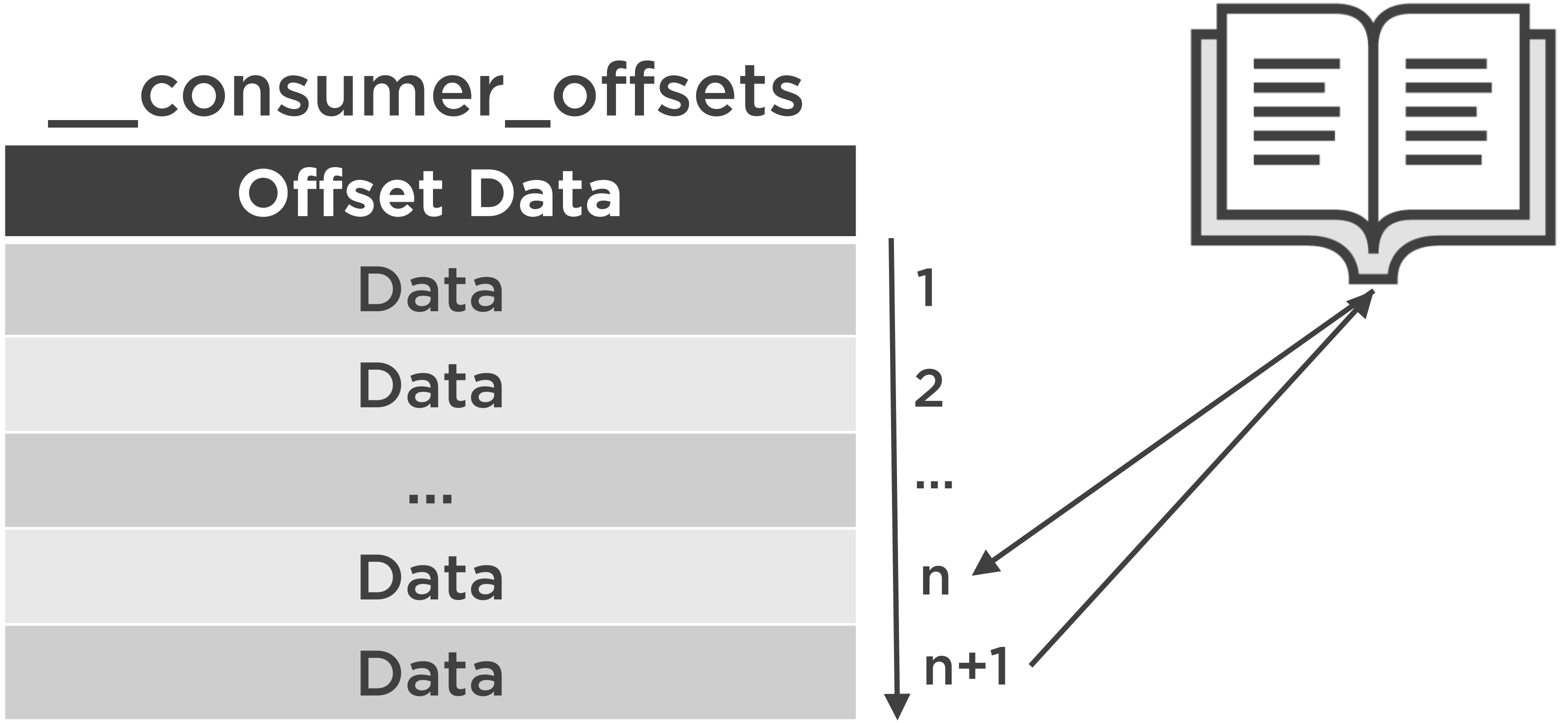
Position: 9

- seek(partitions, position)



5		9	…			51

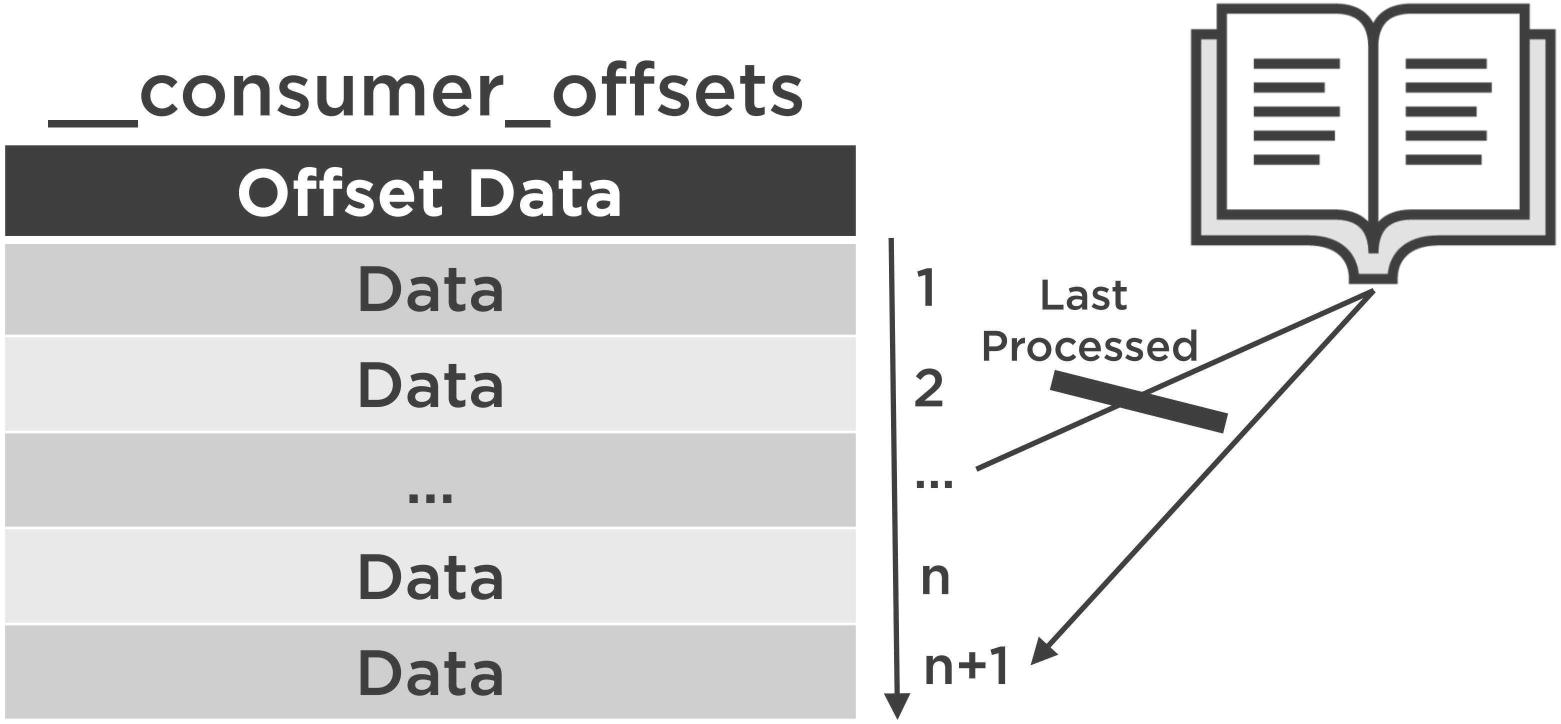
Offset Tracking



__consumer_offsets

Offset Data
Data
Data
...
Data
Data

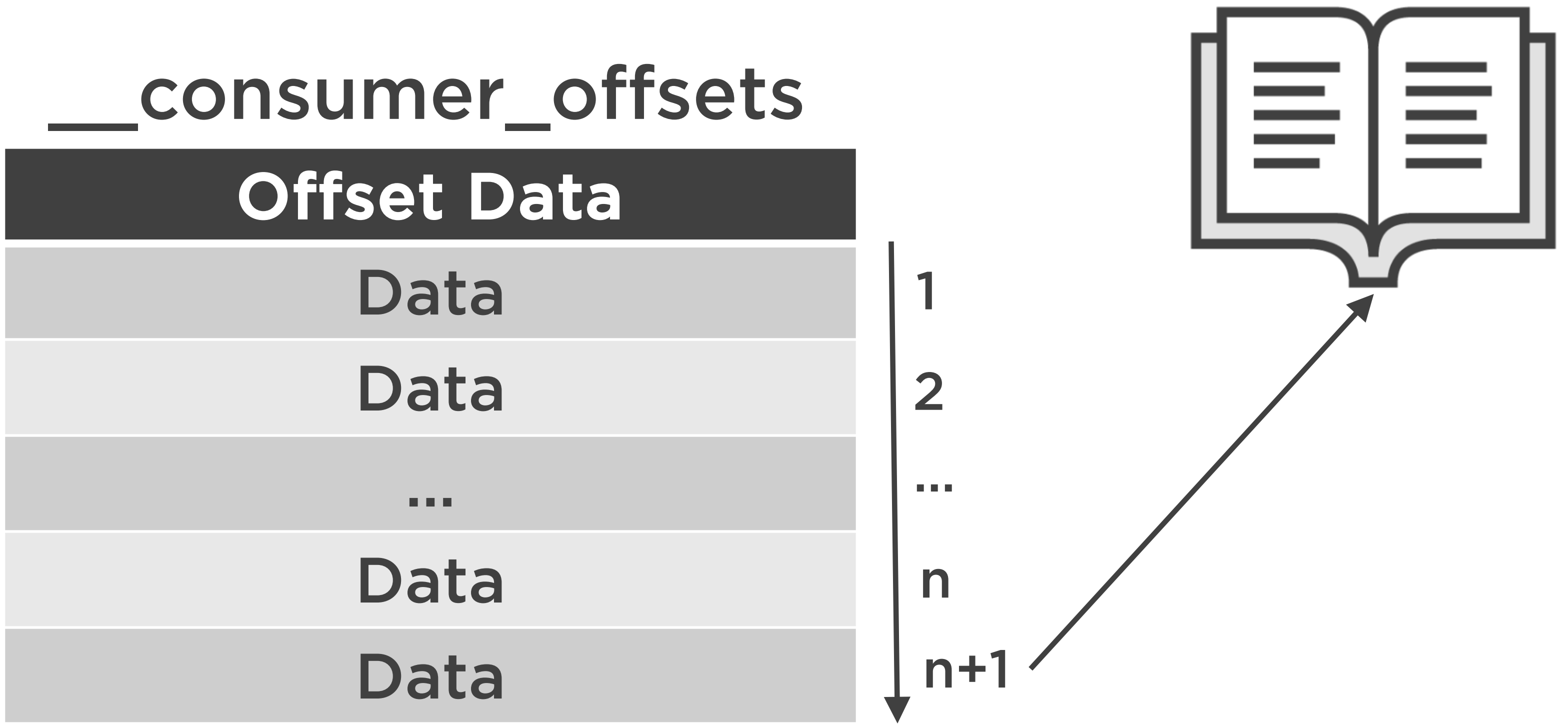
Offset Tracking



__consumer_offsets

Offset Data
Data
Data
...
Data
Data

Offset Tracking

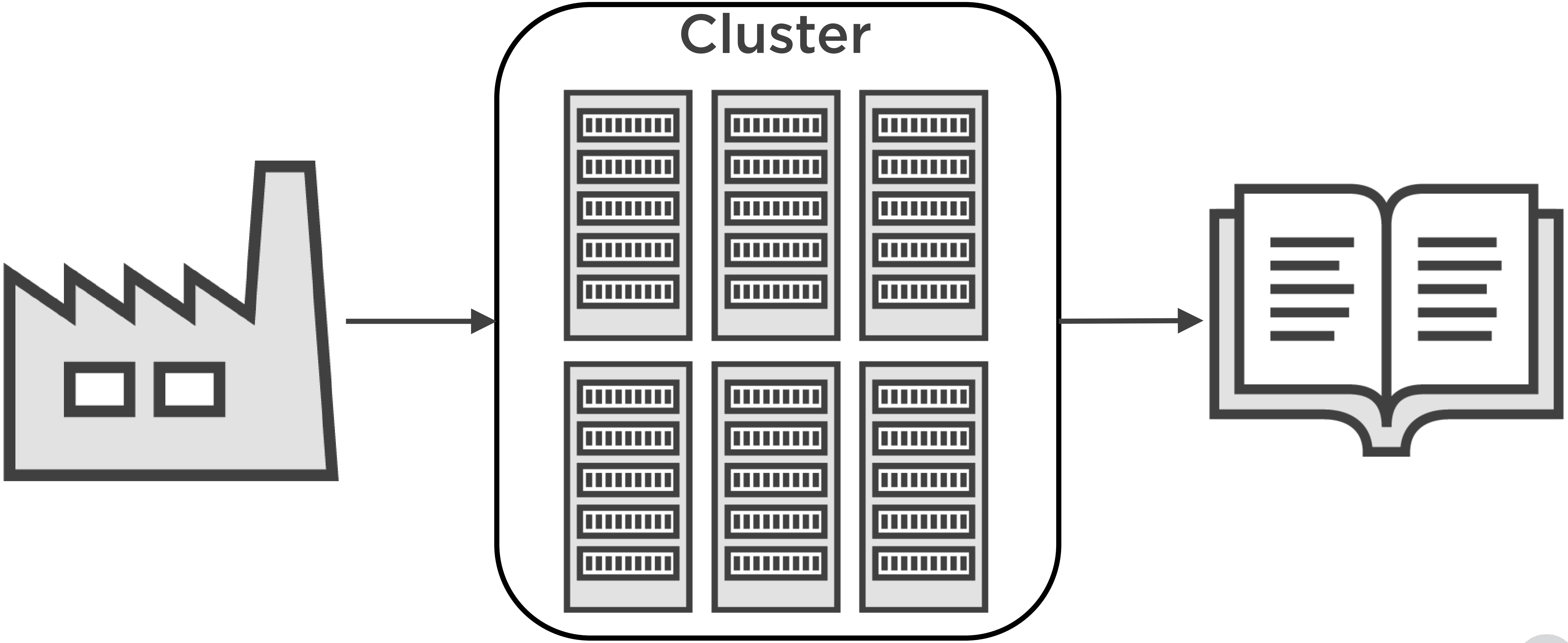


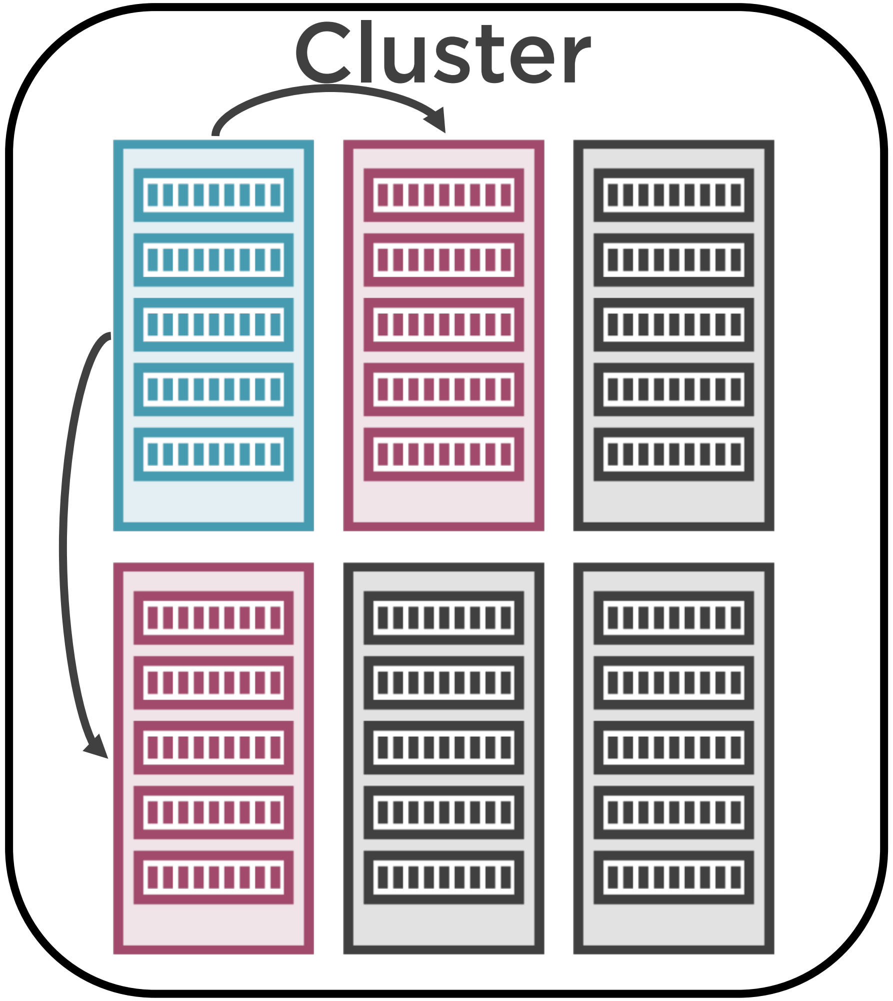
__consumer_offsets

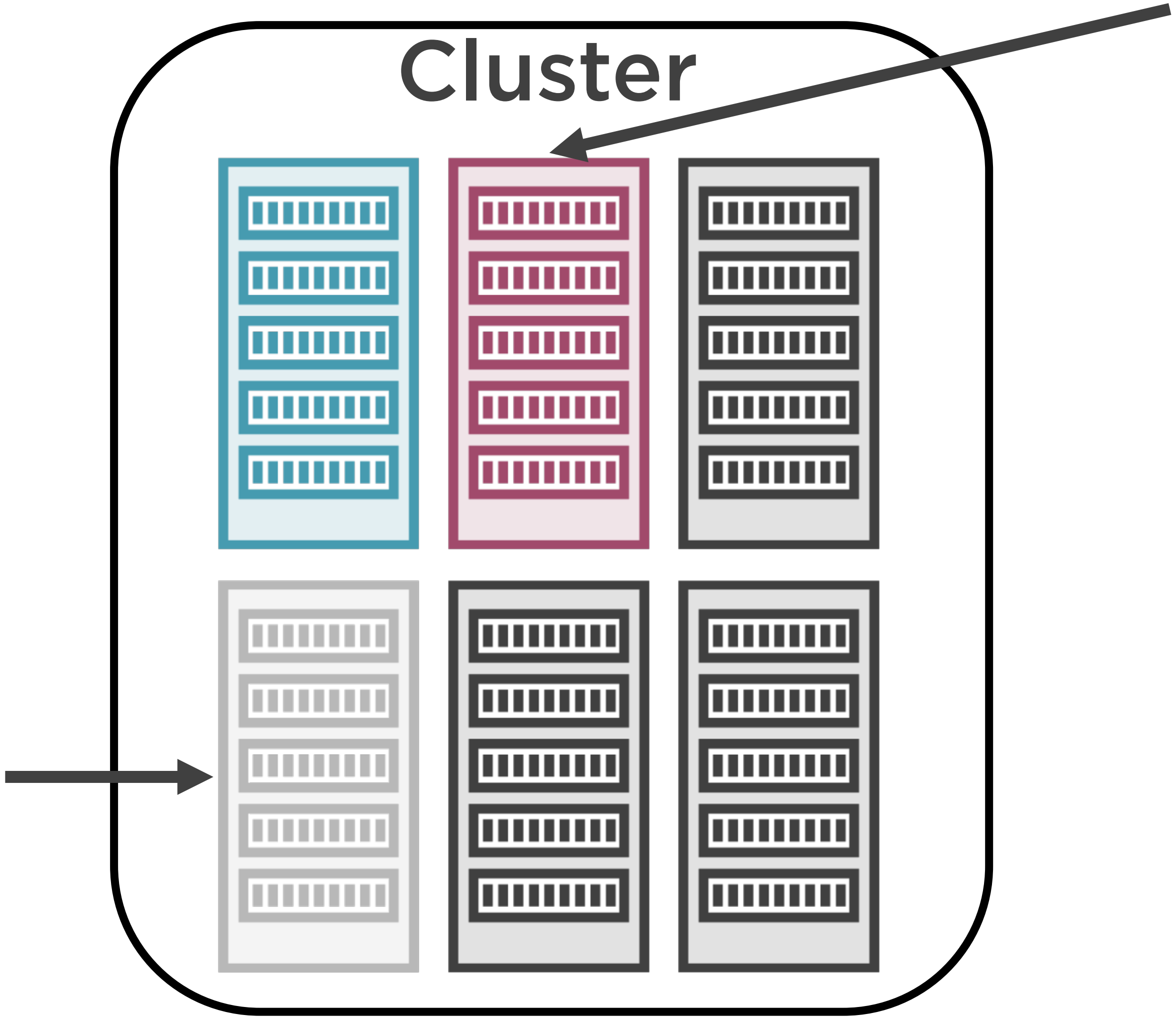
Offset Data
Data
Data
...
Data
Data

Reliability in Production

Reliability




Cluster

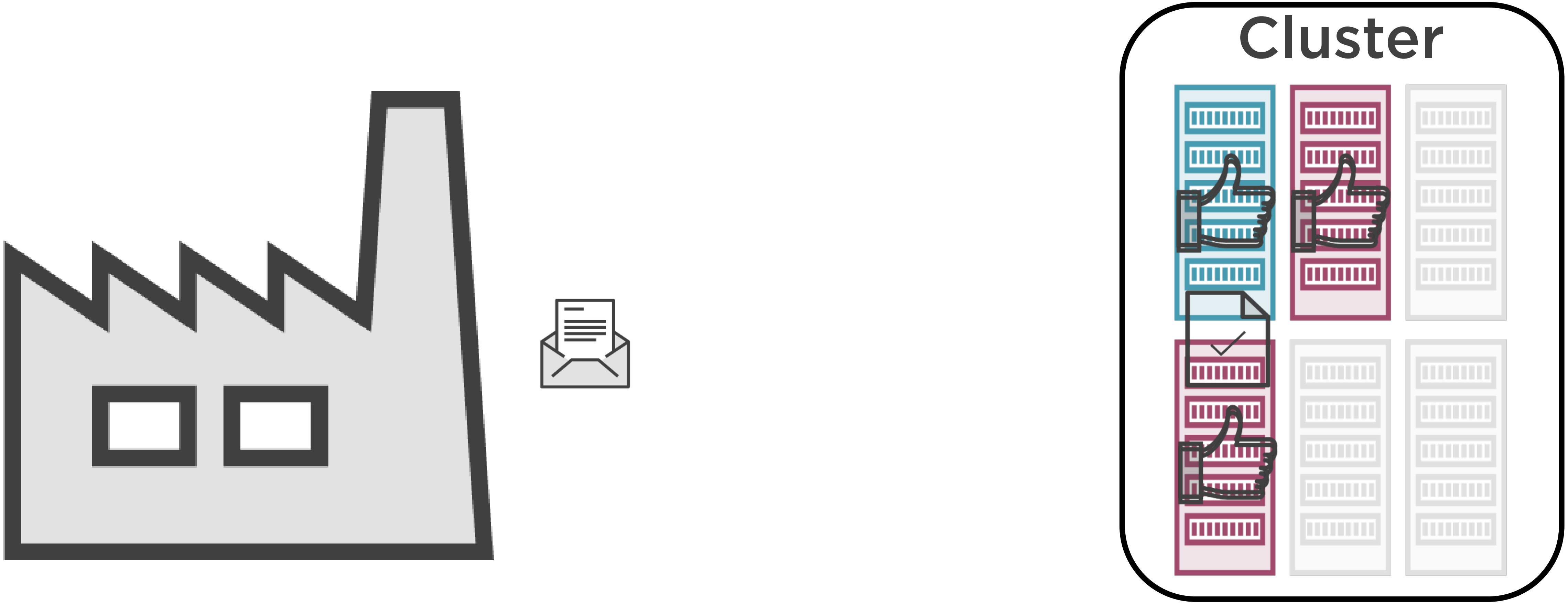


Cluster

In Sync Replica
(ISR)

Out of Sync Replica

acks = all (or -1)



acks = all (or -1)

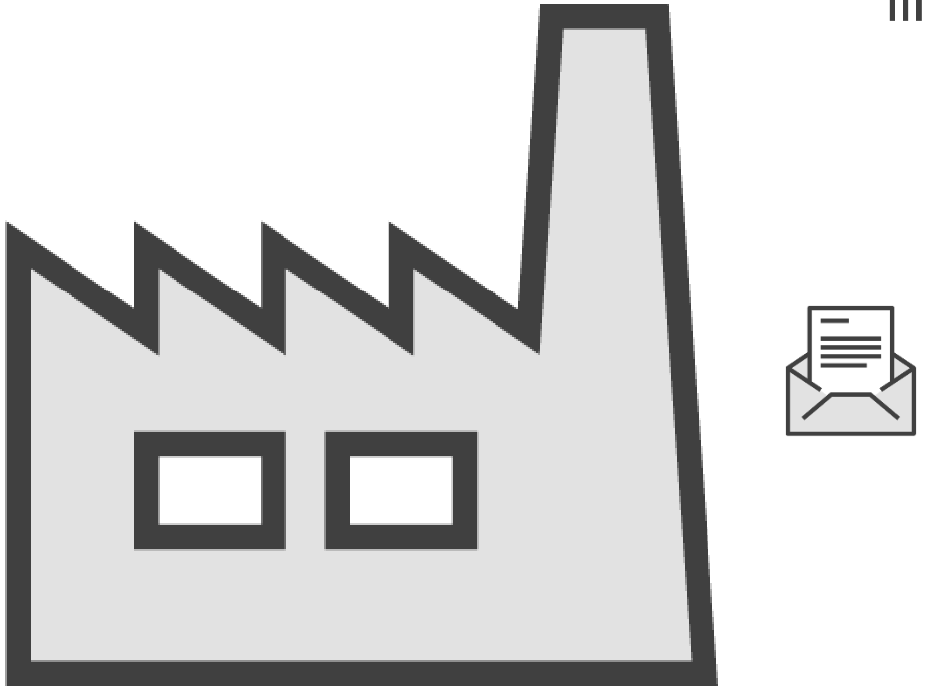
≈ 1 ???

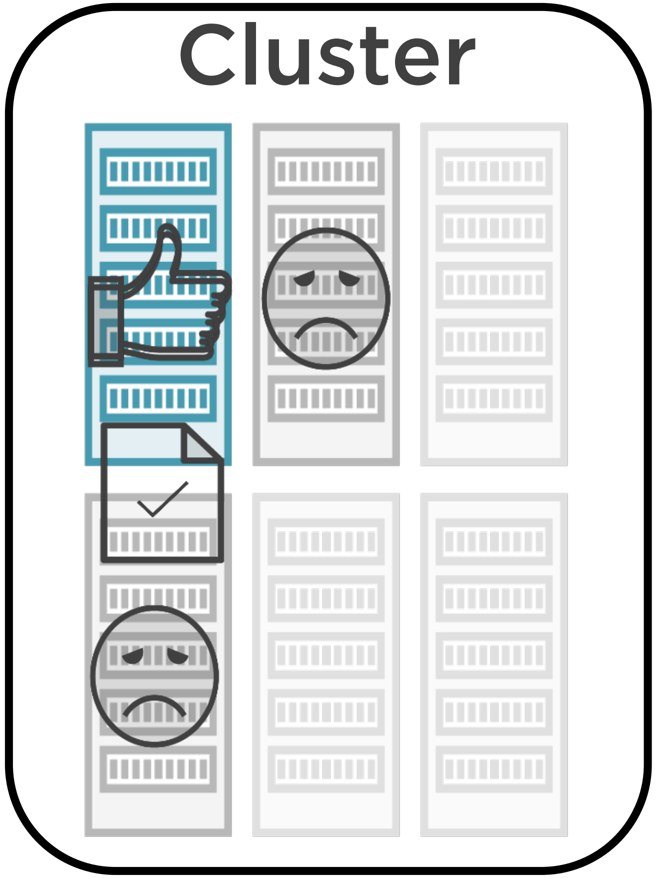
`min.insync.replicas`

`=`

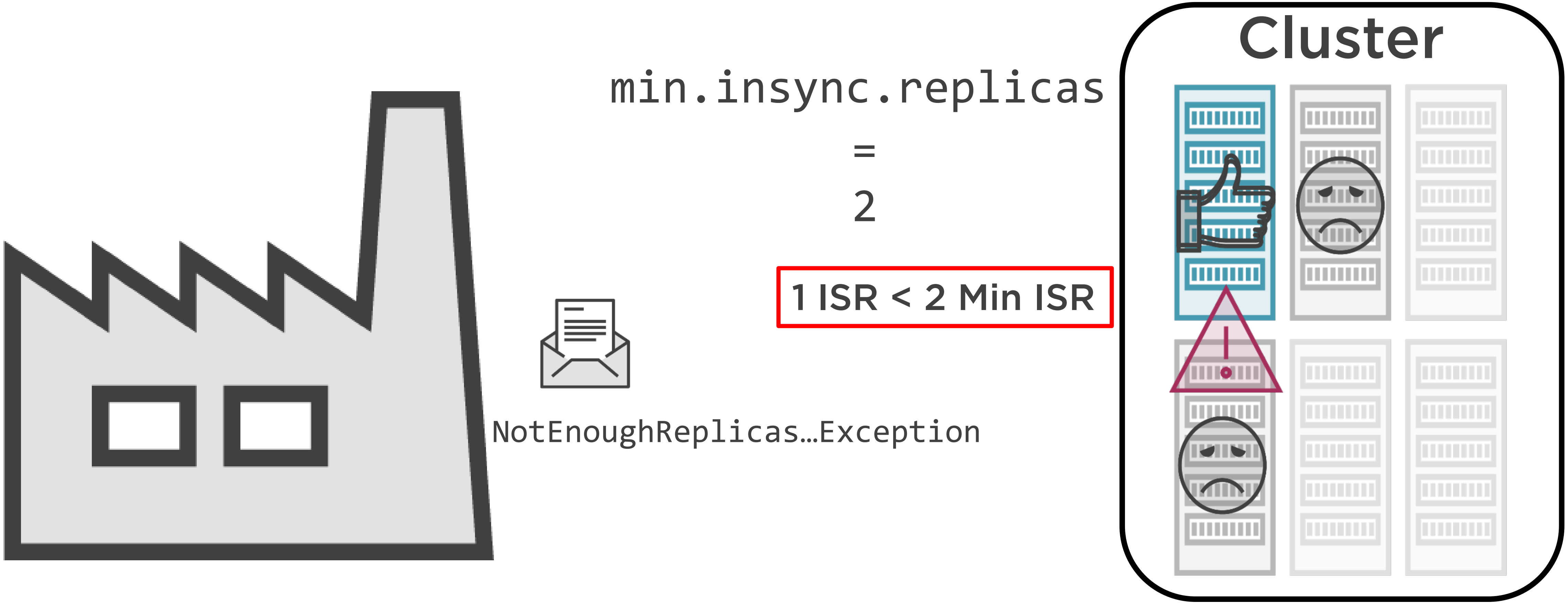
`1`

`(default)`





acks = all (or -1)



min.insync.replicas

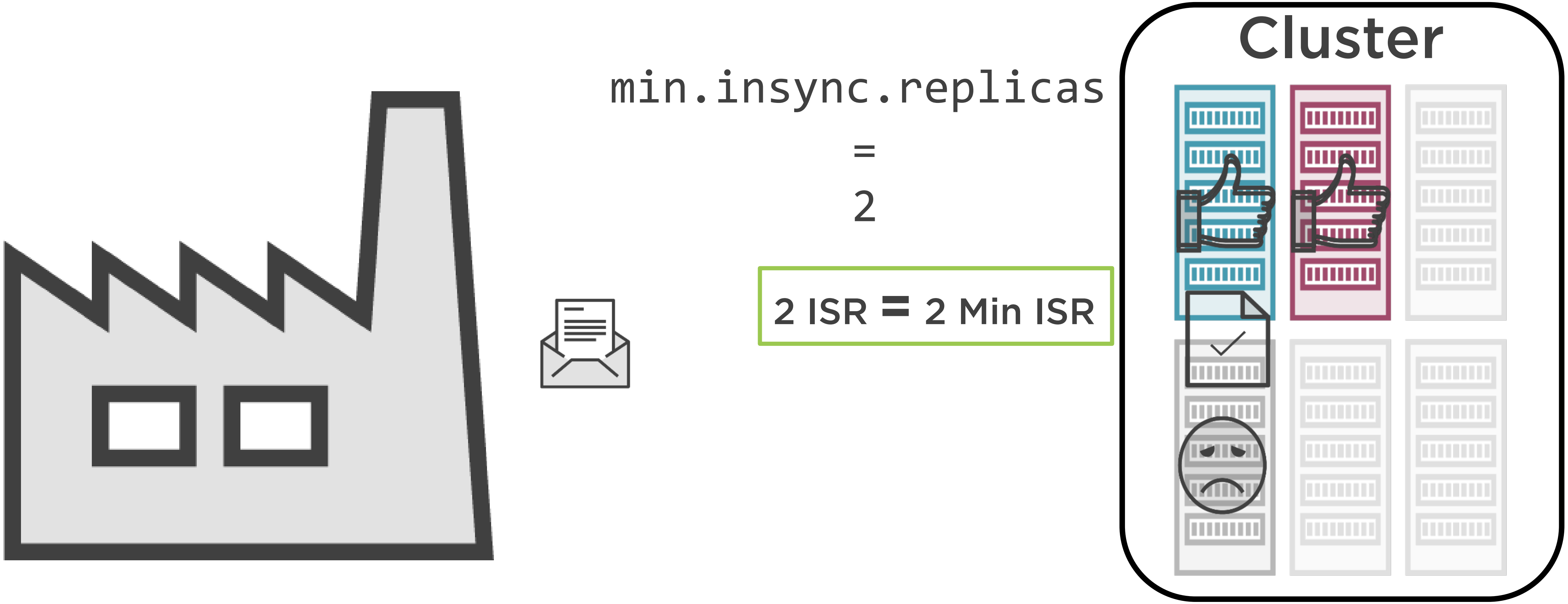
=

2

1 ISR < 2 Min ISR

NotEnoughReplicas…Exception

acks = all (or -1)



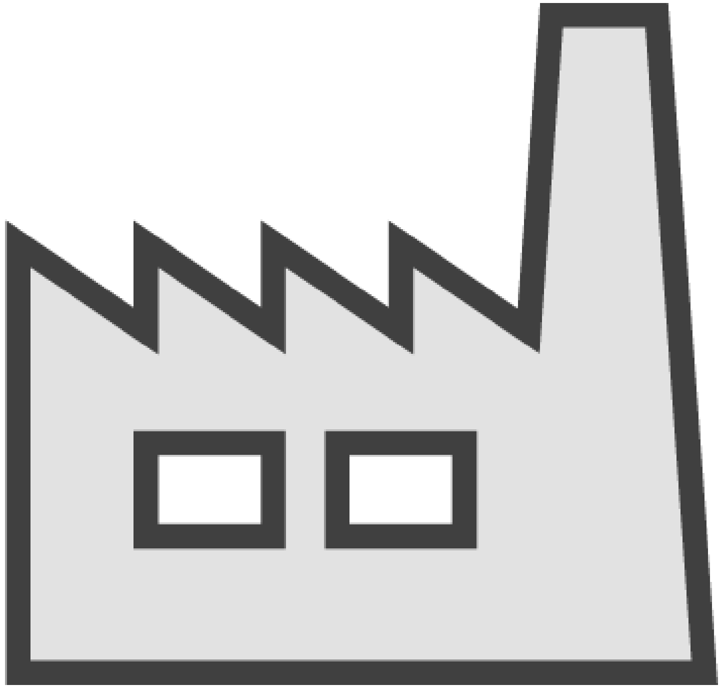
`min.insync.replicas`

=

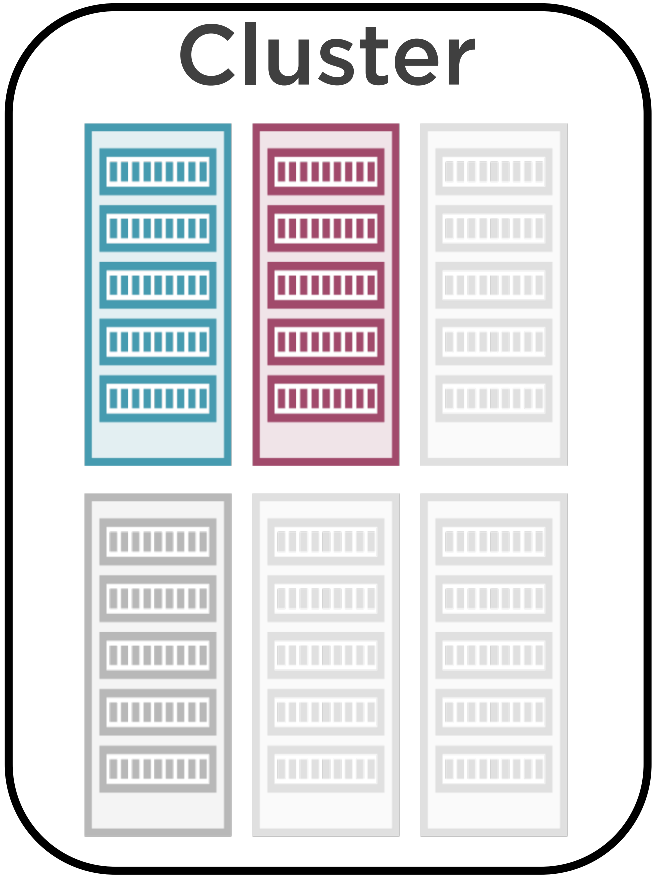
2

2 ISR = 2 Min ISR

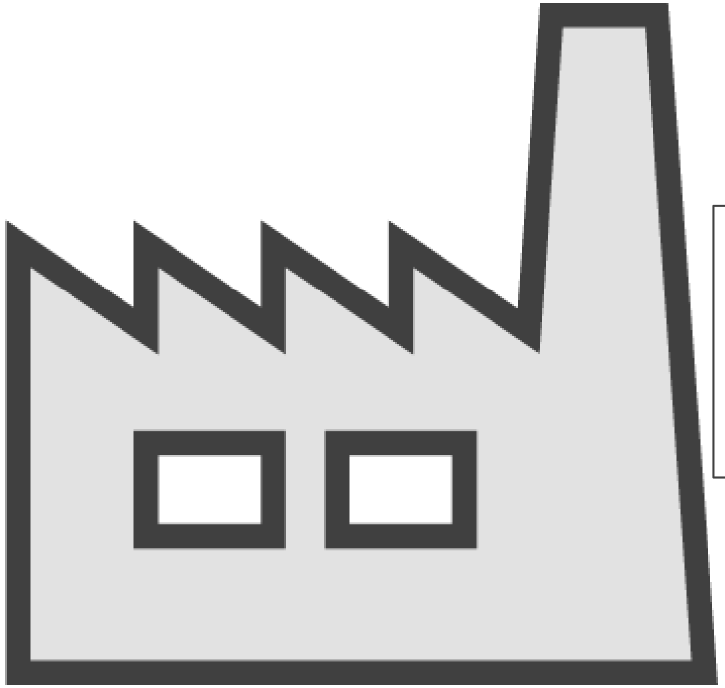
acks = all (or -1)



```
min.insync.replicas
         =
         2
```



acks = all (or -1)

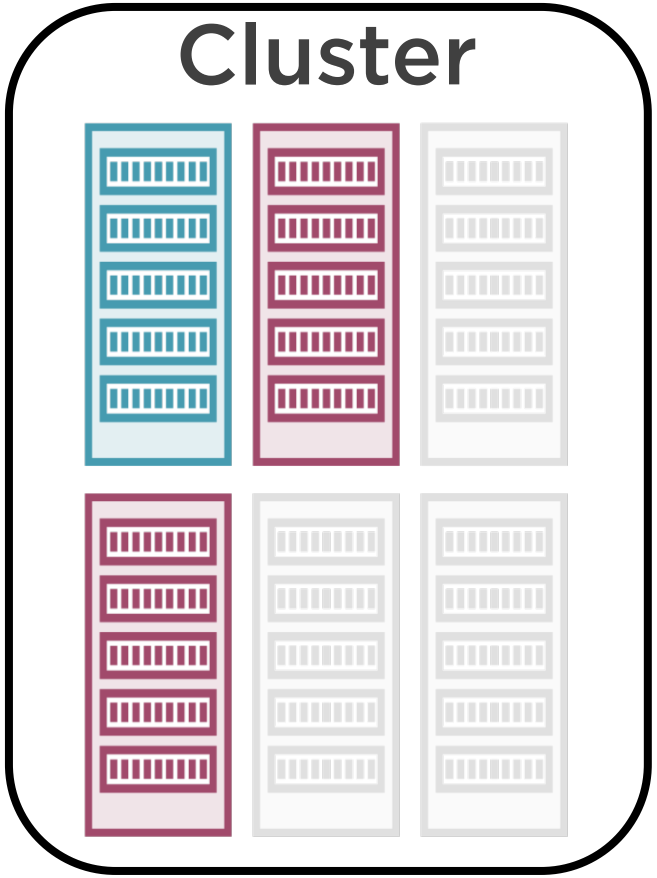


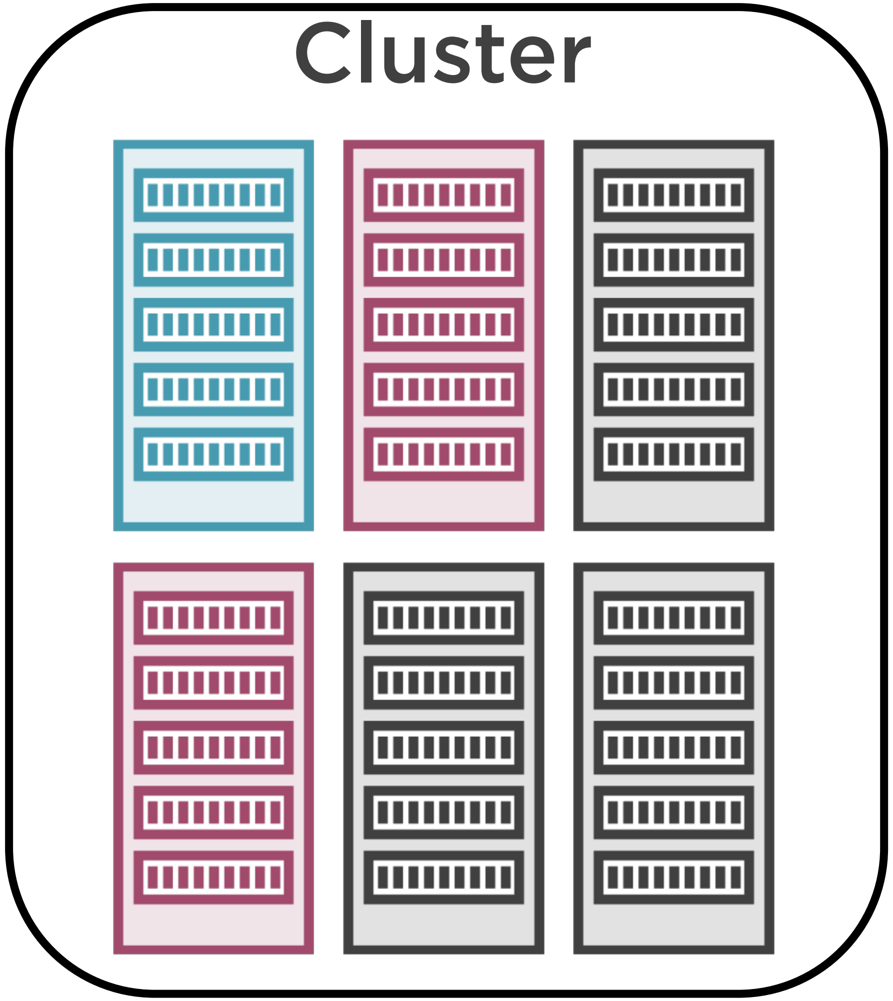
min.insync.replicas

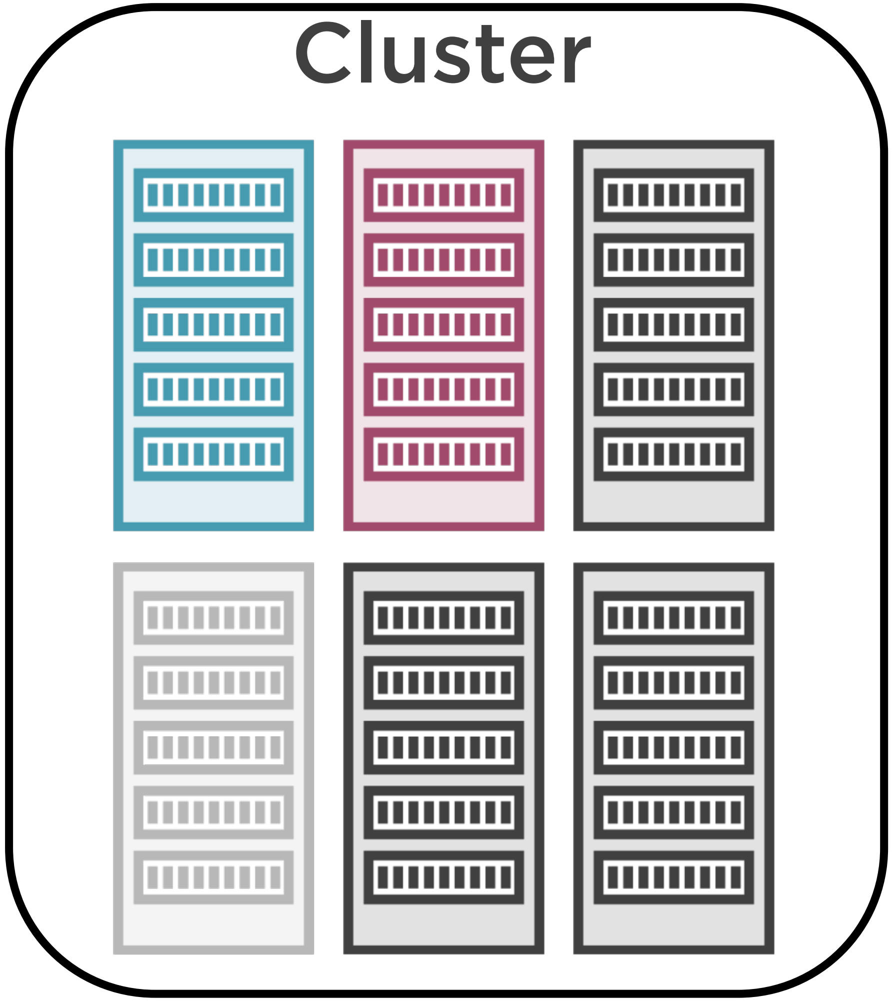
=

2

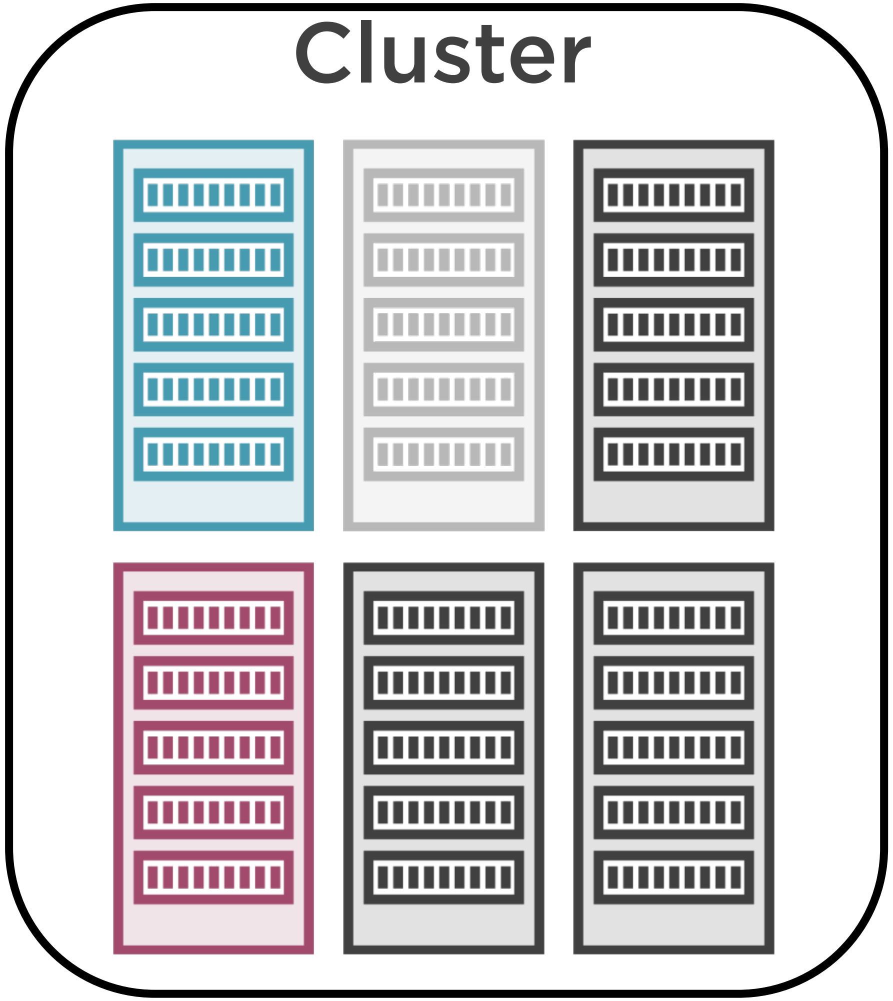
broker = topic default
OR
topic override



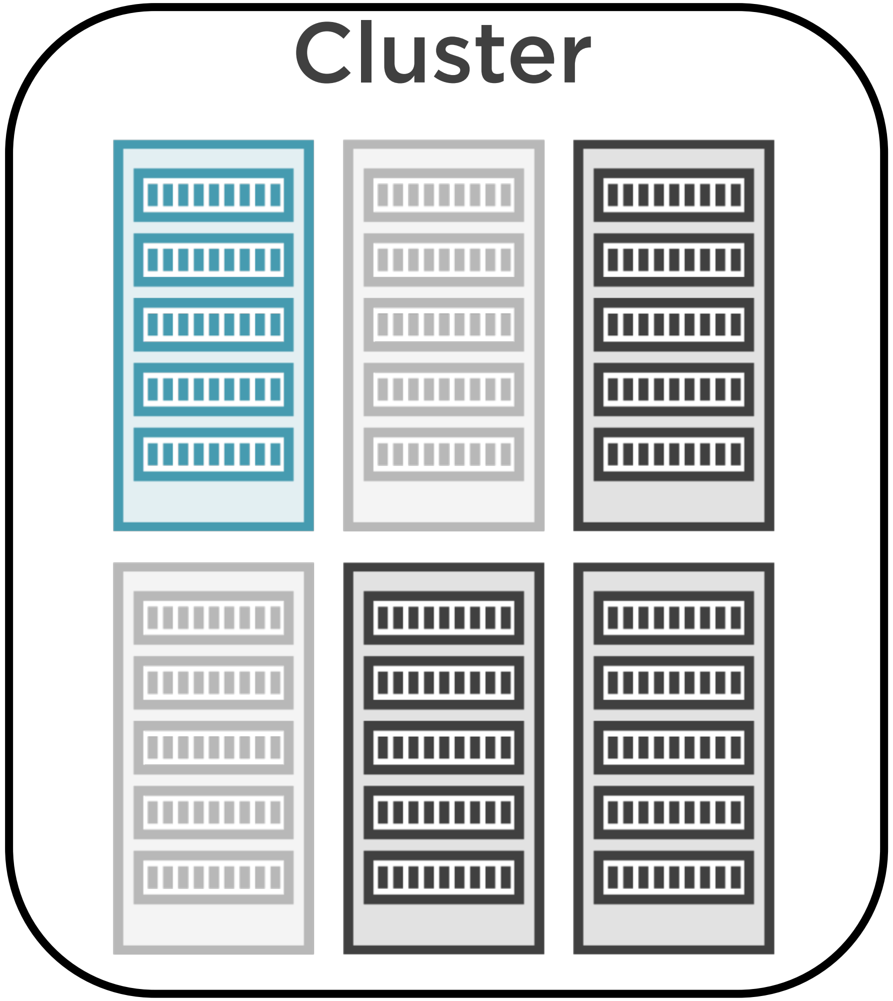

Cluster

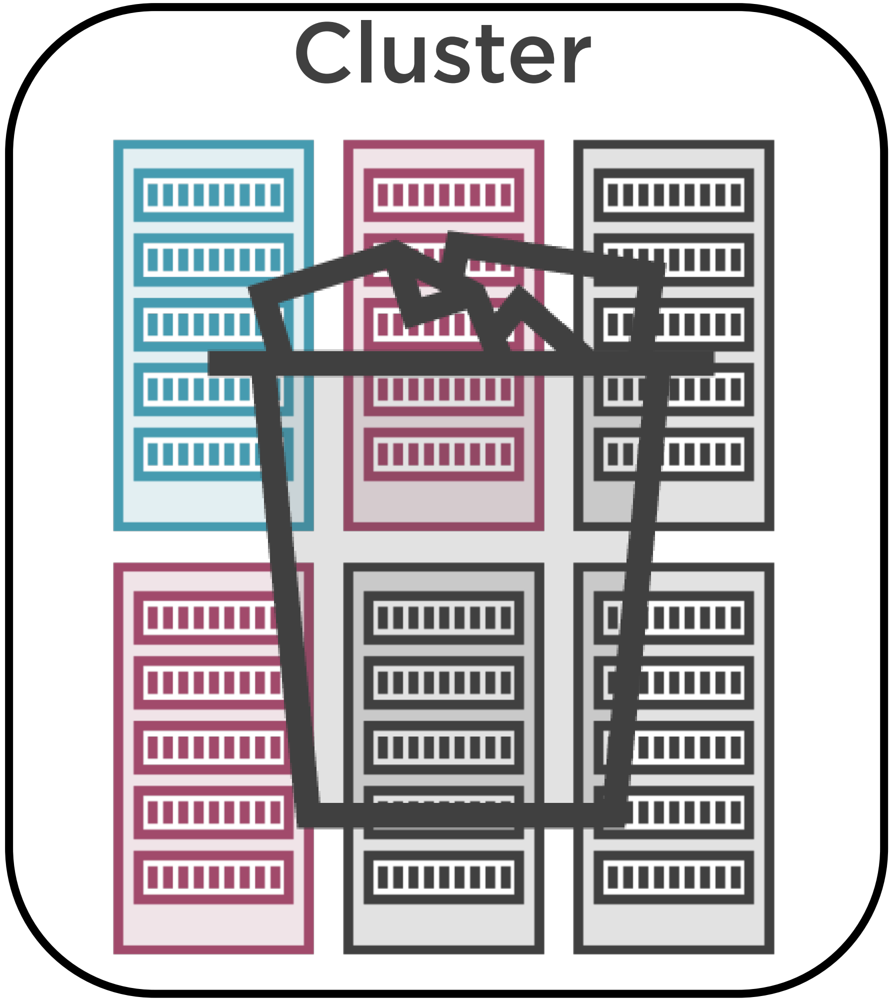


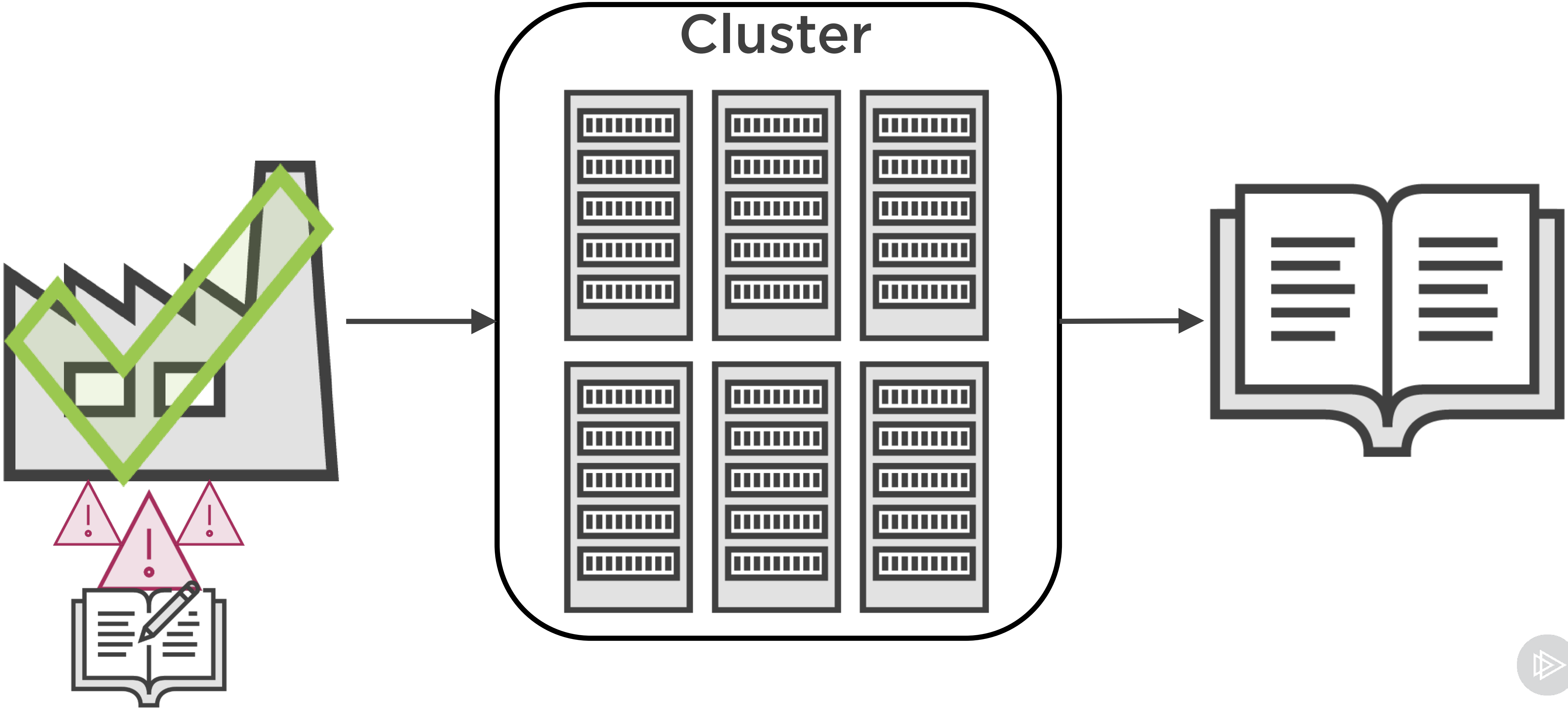
Cluster

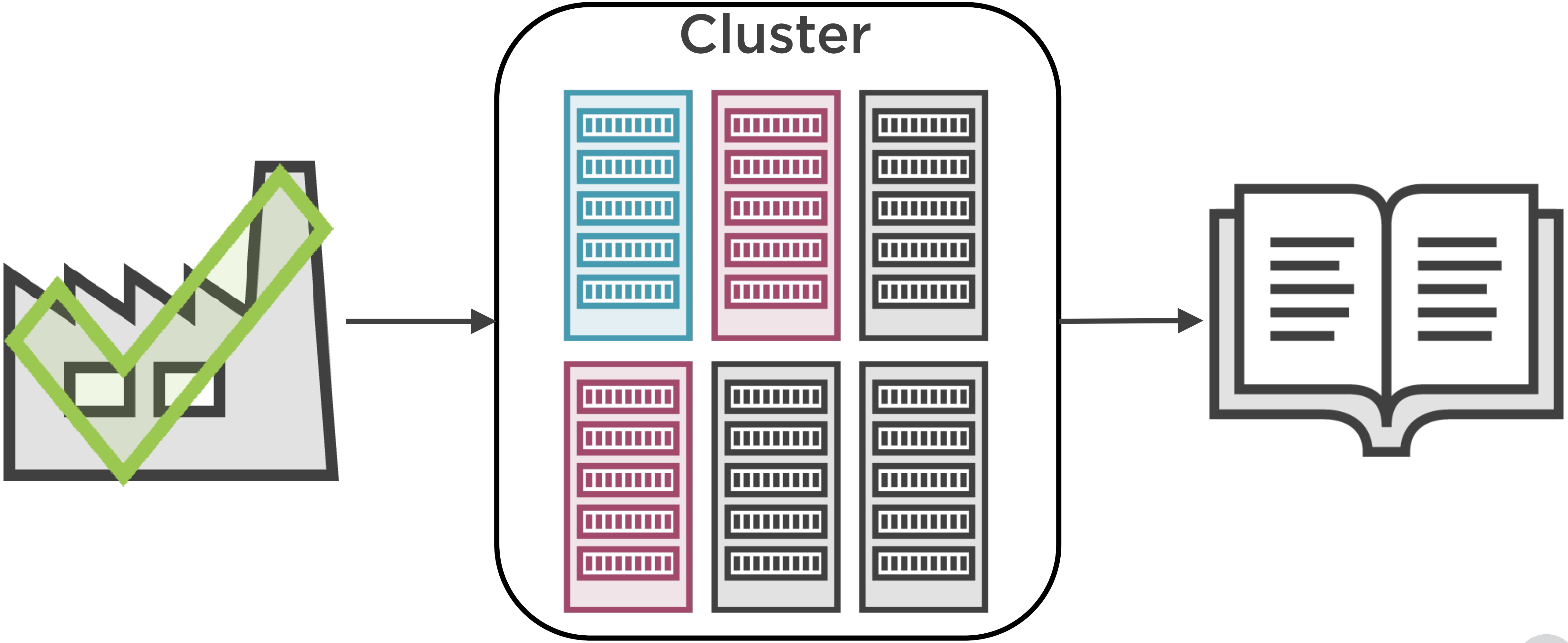


Cluster


Cluster


Cluster


Cluster


Cluster

Offsets

`enable.auto.commit`

`auto.commit.interval.ms`

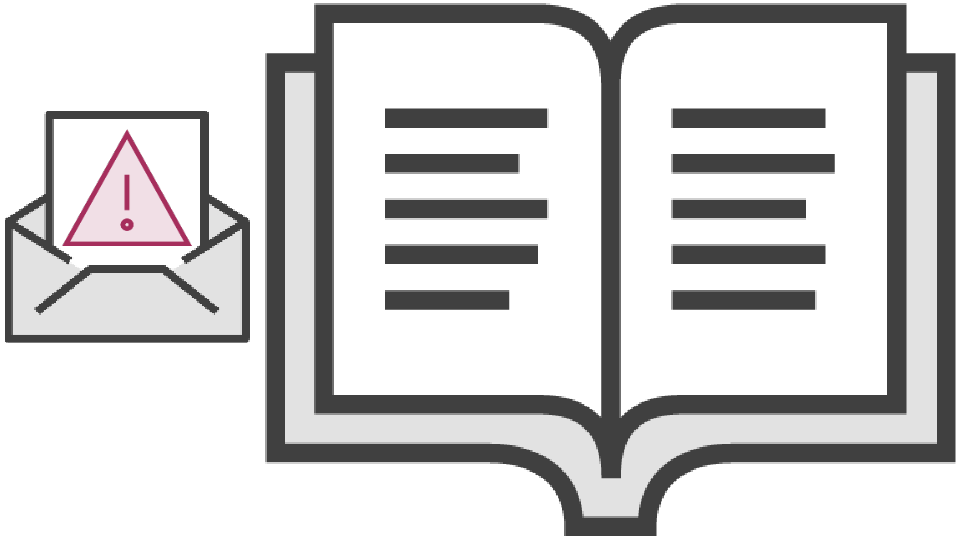
`consumer.commitSync(offsets)`

versus

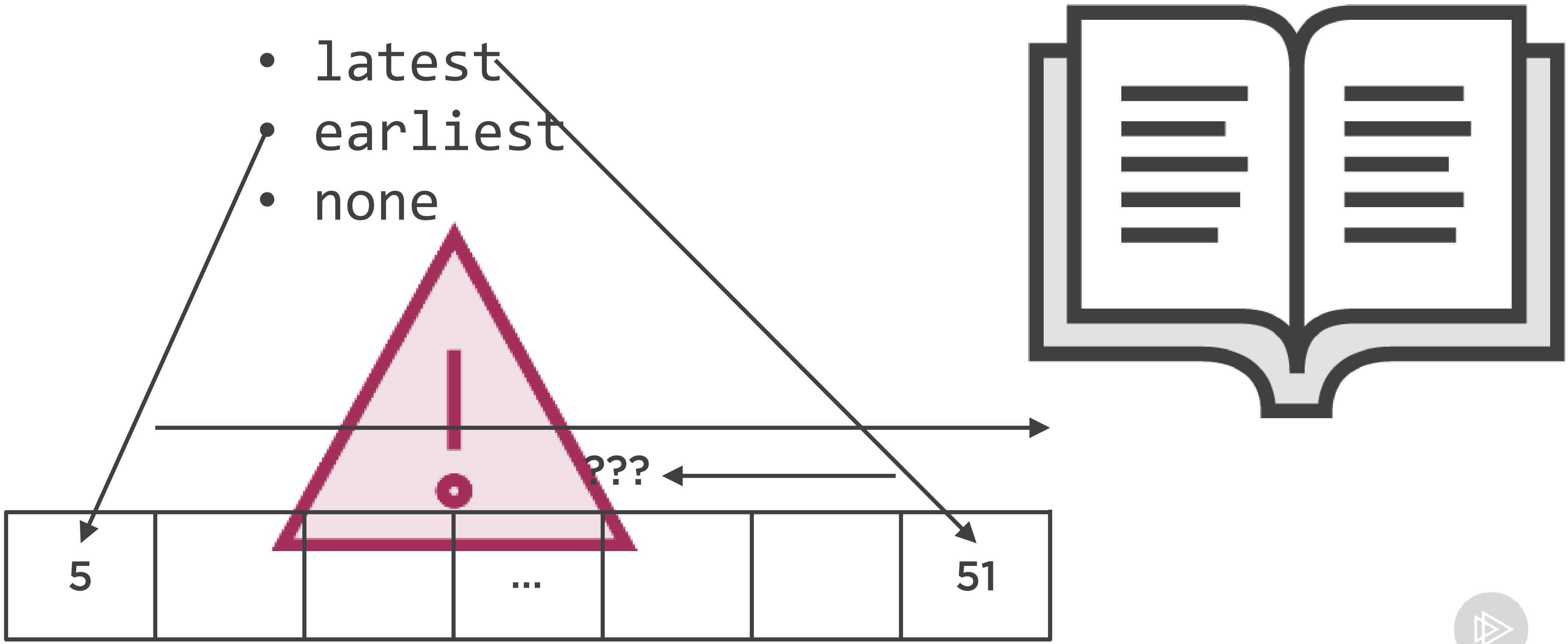
`consumer.commitSync()`

`allOffsets`

Offsets

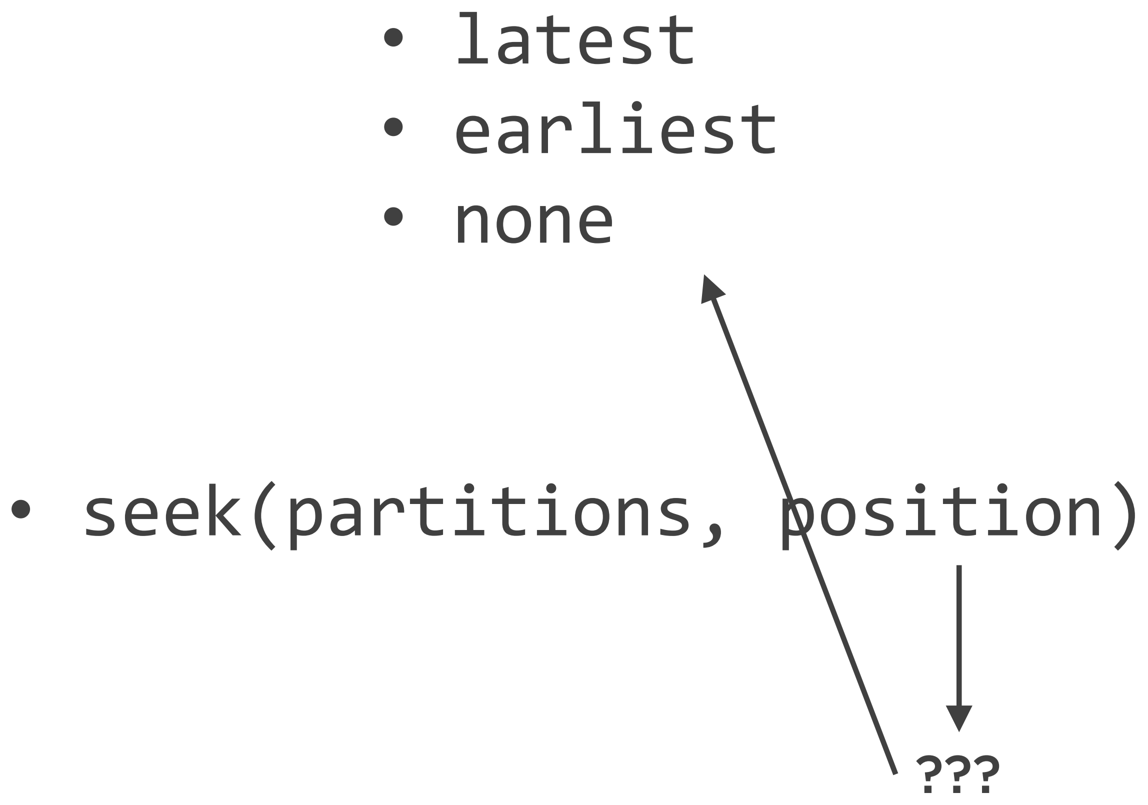


auto.offset.reset



- latest
- earliest
- none

auto.offset.reset



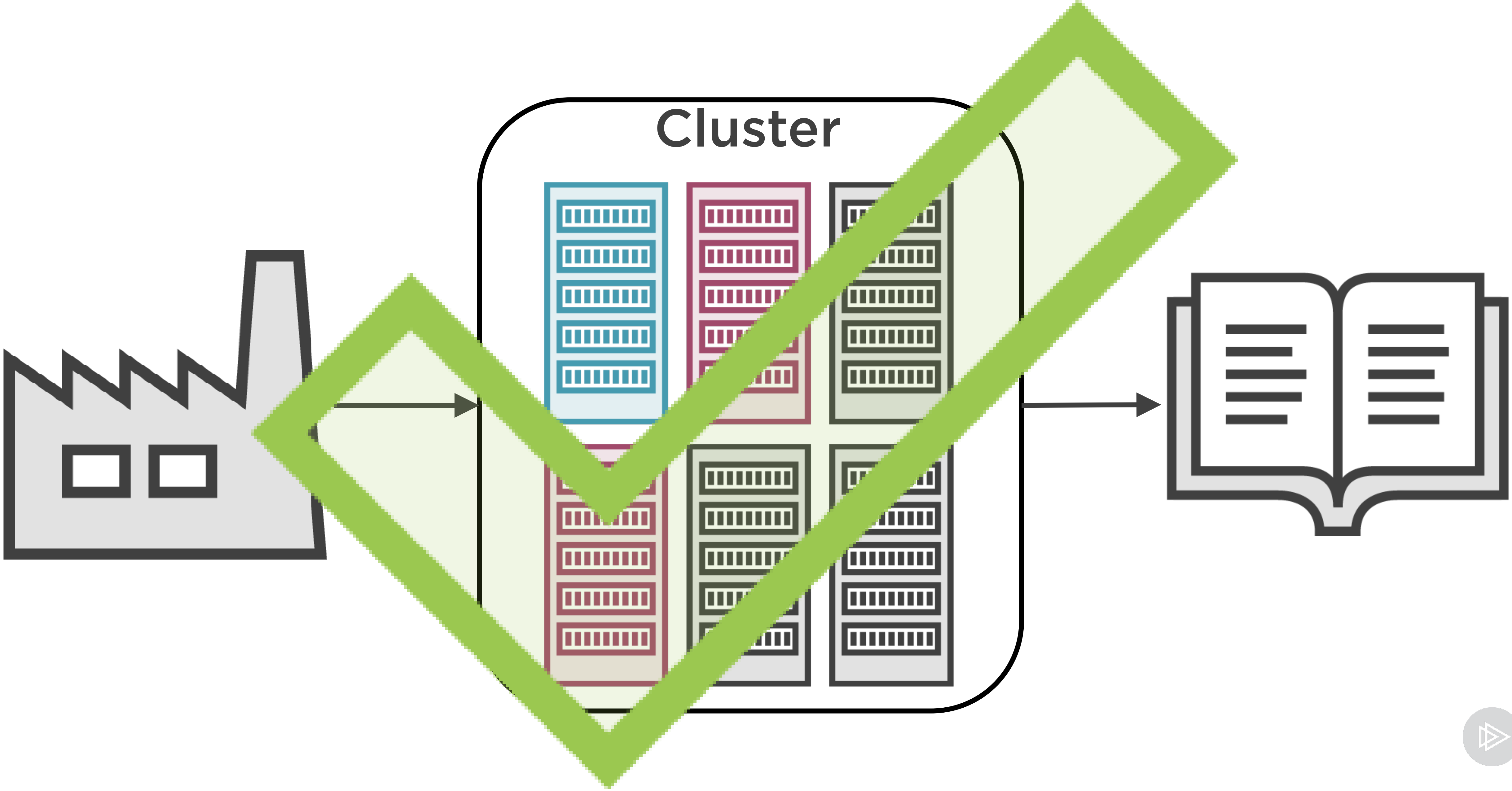
- latest
- earliest
- none

- seek(partitions, position)

???

Reliability Validation

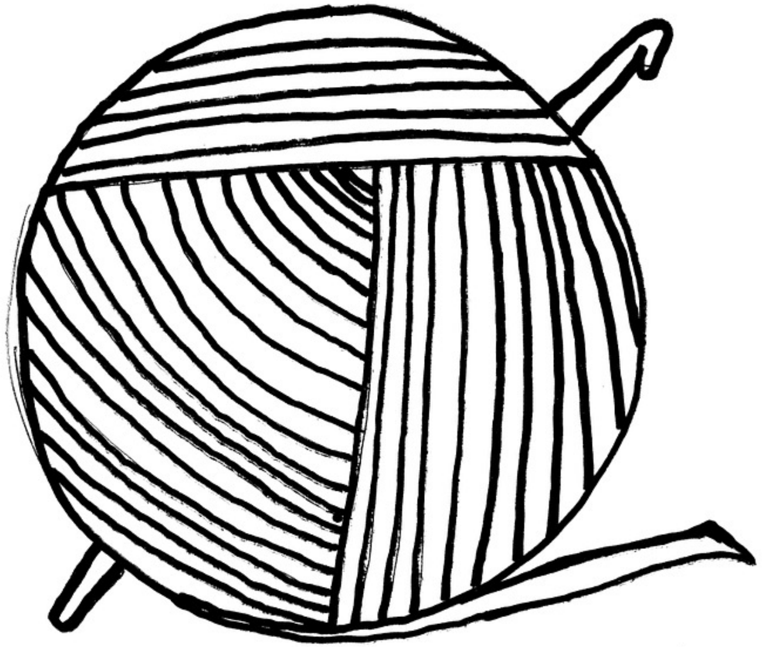
- Reliable Configurations
  - org.apache.kafka.tools.VerifiableProducer
  - org.apache.kafka.tools.VerifiableConsumer
- Test Scenarios
  - Consumer Rebalancing
  - Leader re-election
  - [Consumer/Producer/Broker] rolling restart
- Monitoring!


Cluster

compression.type
=
uncompressed
(default)

- snappy
- lz4
- gzip
- zstd

Empowerment through Custom Messages: Serializers & Deserializers



0 1 1 1 0 1 1 1 1 0

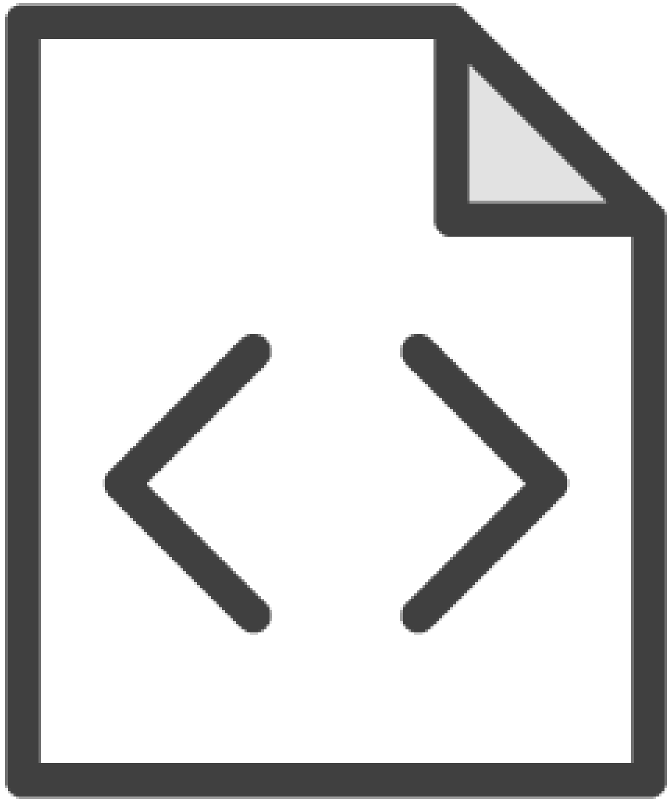
0 0 1 1 🐛 0 0 0 1 1

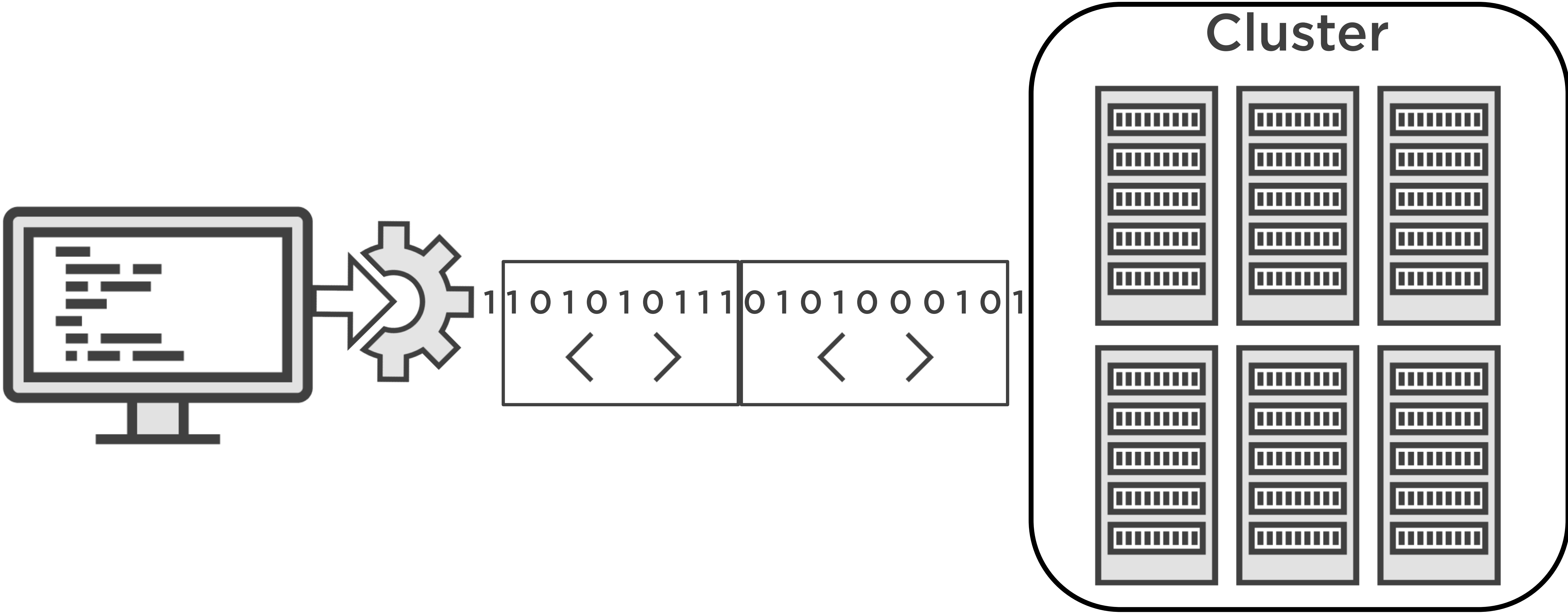
1 0 1 0 1 1 1 0 1 1

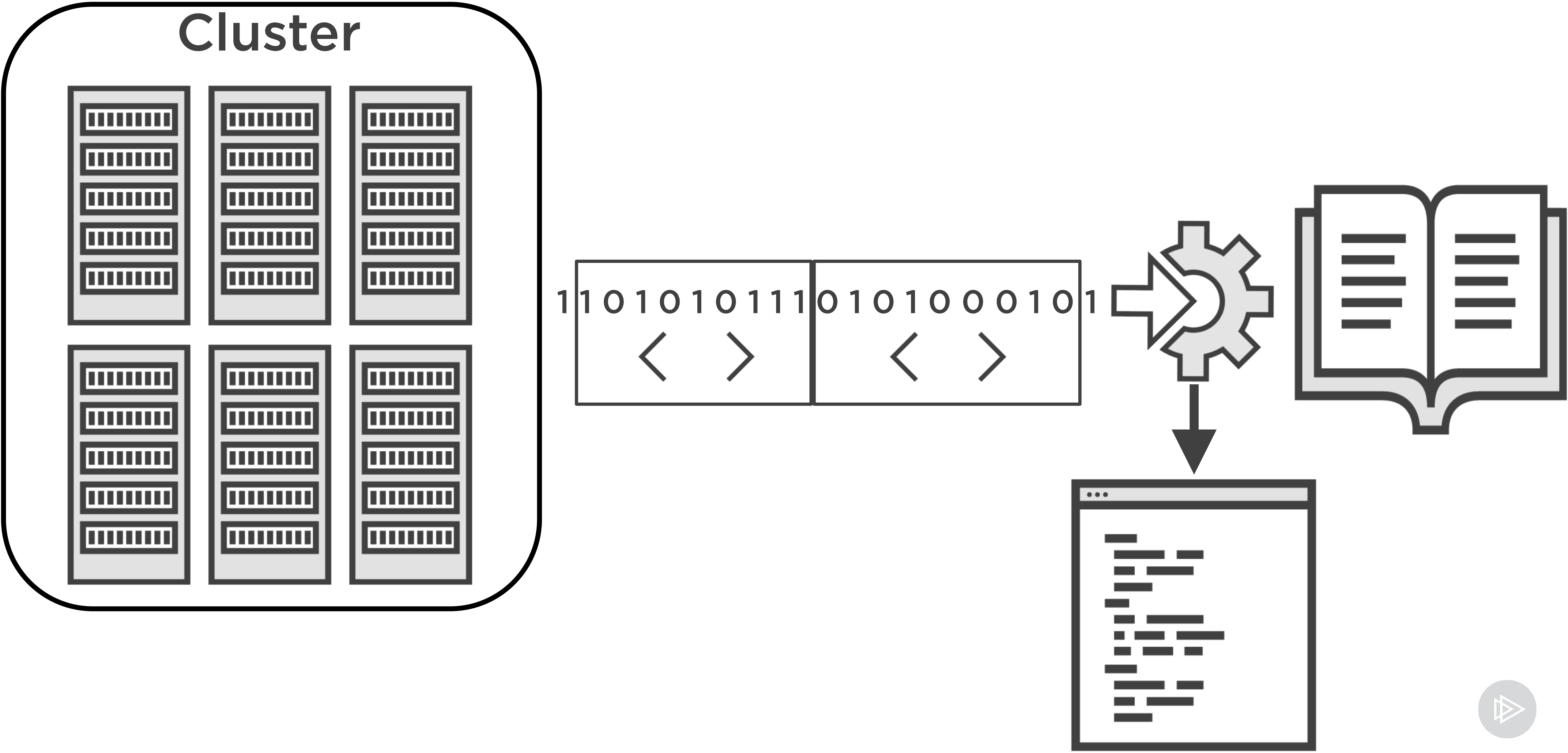
1 0 0 1 0 1 1 1 0 1

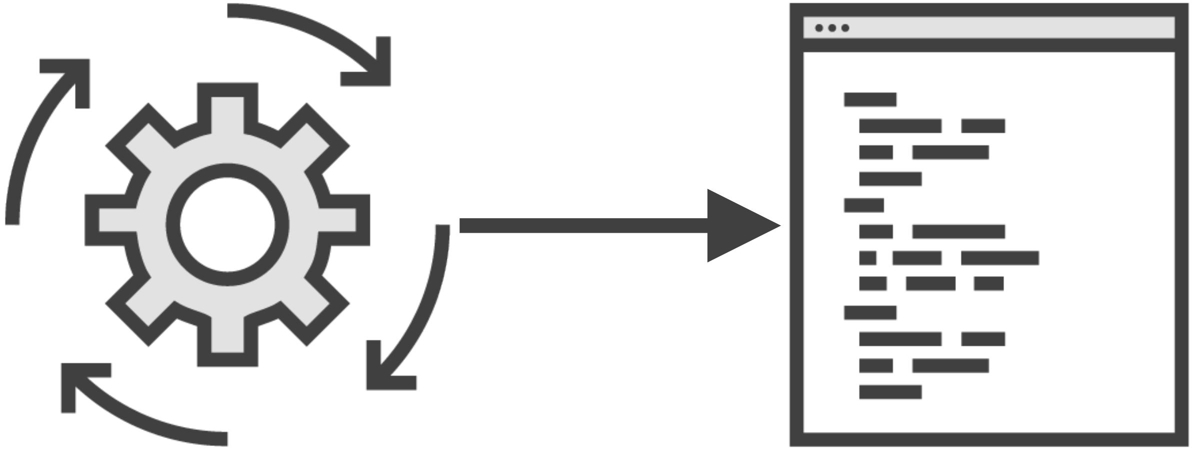
0 1 0 1 0 0 1 1 🐛 0

0 1 1 0 1 0 1 0 1 1

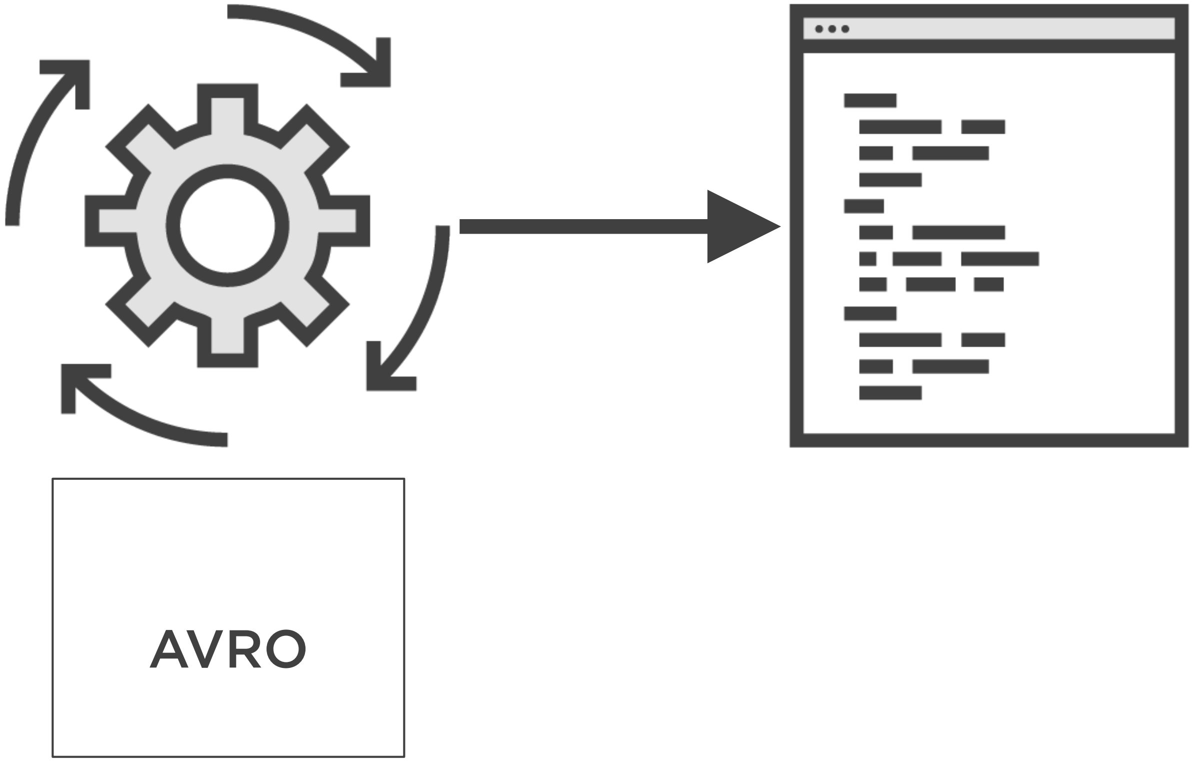



Cluster
1 1 0 1 0 1 0 1 1 1 0 1 0 1 0 0 0 1 0 1

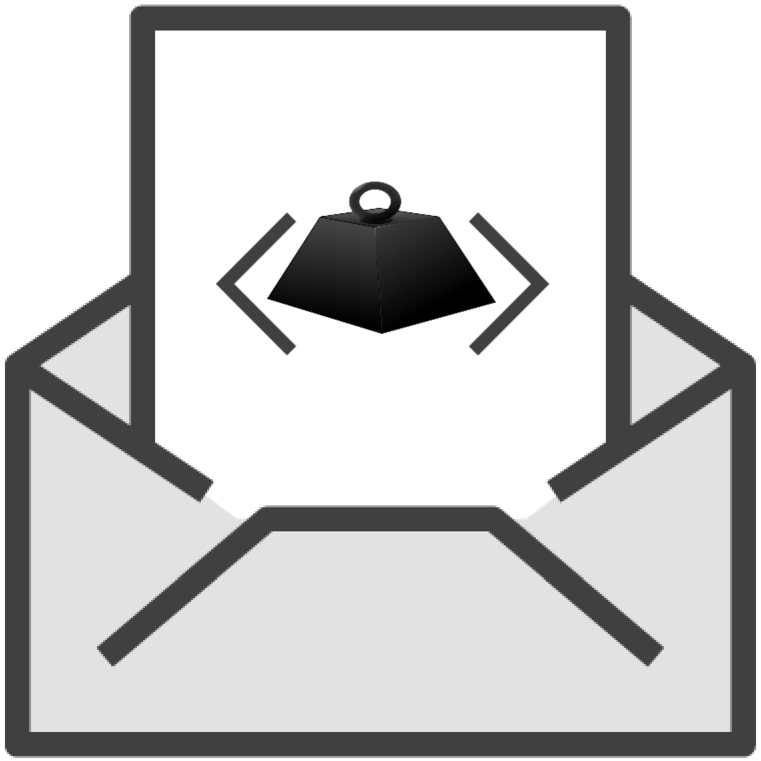

Cluster
1 1 0 1 0 1 0 1 1 1 0 1 0 1 0 0 0 1 0 1
< >
< >



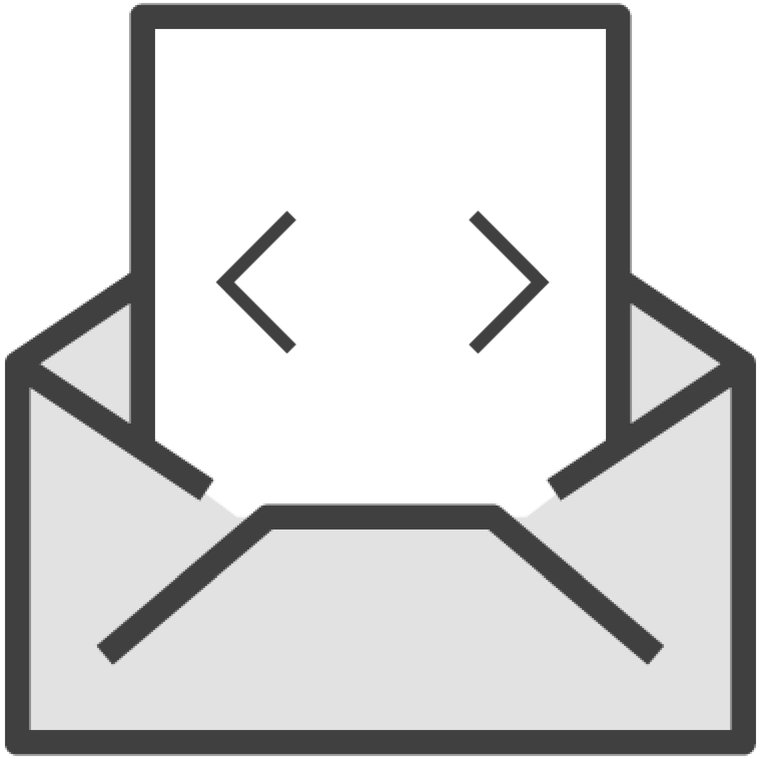
JSON
PROTOBUF
AVRO
...

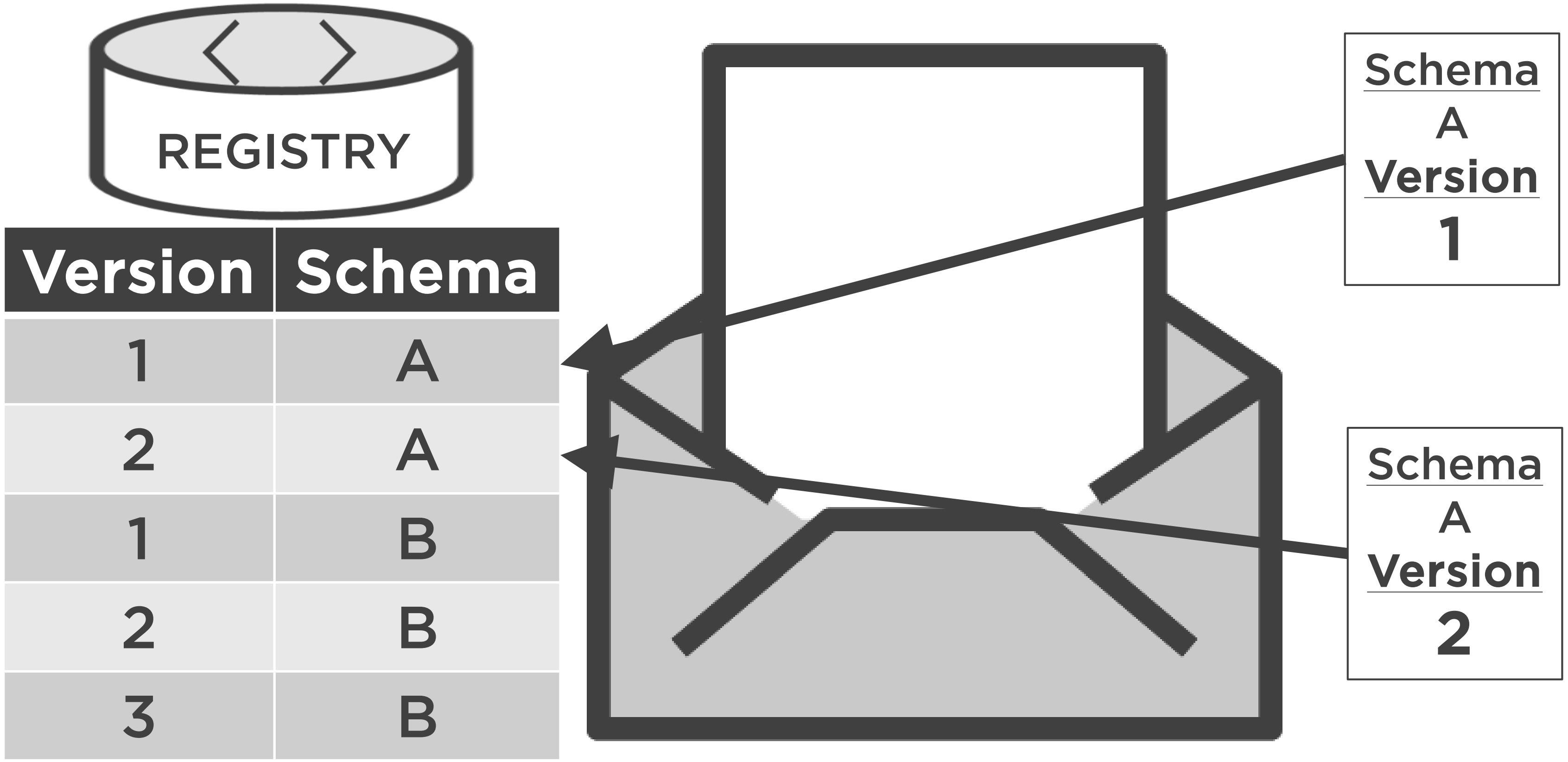


AVRO



Version	Schema
1	A
2	A
1	B
2	B
3	B





REGISTRY

Version	Schema
1	A
2	A
1	B
2	B
3	B

Schema A Version 1

Schema A Version 2

Resources

- Configuring Apache Kafka for Performance and Resource Management
  - cloudera.com/documentation/kafka/latest/topics/kafka_performance.html
- Running Kafka in Production
  - docs.confluent.io/current/kafka/deployment.html
- Decoupling Systems with Apache Kafka, Schema Registry and Avro
  - confluent.io/blog/decoupling-systems-with-apache-kafka-schema-registry-and-avro
- Avro Documentation
  - avro.apache.org/docs/current
- Schema Registry Installation Documentation
  - docs.confluent.io/current/schema-registry/docs/installation.html#development

Summary

Configuration

- Producer
- Consumer

Reliability

Message Typing via Avro